

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mirko Rastović

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

doc. dr. sc. Tomislav Stipančić

Student:

Mirko Rastović

Zagreb, 2017.

Ovim izjavljujem da sam ovaj rad radio samostalno, služeći se znanjem stečenim tijekom studija i koristeći navedenu literaturu.

Želio bih zahvaliti:

Voditelju rada doc.dr.sc. Tomislavu Stipančiću na vrijednim savjetima vezanim za izradu rada, uz poseban naglasak na ukazanu pomoć i strpljenje vezanim za moju situaciju pred kraj izrade rada.

Zahvaljujem i obitelji na podršci i razumijevanju tijekom studiranja, mnogim prijateljima, kao i kolegama koji su mi pomagali tijekom studija i izrade ovog rada, a naročito kolegi Nikoli Korenu.

Posebno zahvaljujem Deji na neizmjernoj potpori i razumijevanju tijekom trajanja mog diplomskog studija, a pogotovo na svojoj podršci i pomoći tijekom svakodnevnih teških trenutaka posljednjih 18 mjeseci uzrokovanih mojom bolešću.

Mirko Rastović



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur. broj:	

DIPLOMSKI ZADATAK

Student: **Mirko Rastović**

Mat. br.: 0035174140

Naslov rada na
hrvatskom jeziku:

**Sinkronizirano upravljanje IP PTZ kamerom temeljeno na
semantičkim mrežnim uslugama**

Naslov rada na
engleskom jeziku:

**Synchronized Control of a IP PTZ Camera based on Semantic Web
Services**

Opis zadatka:

Semantičke mrežne tehnologije se temelje na mrežnom komunikacijskom protokolu SOAP (*Simple Object Access Protocol*) te zapisnicima definiranim u programskom jeziku WSDL (*Web Services Description Language*) koji sadržavaju definicije i opise funkcionalnosti uređaja na mreži. Uređaji čiji je rad temeljen na semantičkim mrežnim uslugama mogu s manjim ili većim stupnjem autonomnosti dijeliti dostupne informacije te izvršavati željene aktivnosti kako u virtualnom, tako i u realnom svijetu. Protokoli semantičkih mrežnih tehnologija pritom moraju omogućiti siguran i pravovremen prijenos informacija.

U zadatku je potrebno:

- razviti programsku podršku koja osigurava komunikaciju između računala tražitelja usluge i mrežne kamere (oznaka: IP) s tri stupnja slobode gibanja (oznaka: PTZ - *Pan Tilt Zoom*) temeljenu na semantičkim mrežnim uslugama, poštujući pritom zahtjeve u pogledu sigurnog prijenosa podataka i privatnosti
- odrediti minimalni broj točaka nadzora unutar stvarnog prostora koje osiguravaju IP PTZ kameri da s dovoljnom razinom detaljnosti pregleda zadani radni prostor.

Programsko rješenje je potrebno temeljiti na programskom paketu WAMP (skraćenica: *Windows – Apache – MySQL – PHP*) koji se koristi na *Windows* računalnom operativnom sustavu te uključuje *Apache* poslužitelj web-stranica, bazu podataka *MySQL* i podršku za programski jezik *PHP*.

Razvijenu programsku podršku potrebno je eksperimentalno verificirati na opremi dostupnoj u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava koja uključuje IP PTZ kameru i upravljačko računalo sa svom potrebnom mrežnom opremom.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

11. svibnja 2017.

Rok predaje rada:

13. srpnja 2017.

Predviđeni datum obrane:

19., 20. i 21. srpnja 2017.

Zadatak zadao:

Doc. dr. sc. Tomislav Stipančić

Predsjednica Povjerenstva:

Prof. dr. sc. Biserka Runje

SADRŽAJ

1.	UVOD	1
2.	INTERNET	3
2.1.	POVIJEST INTERNETA	3
2.2.	KLASIČNI <i>WEB</i>	8
2.3.	SEMANTIČKI <i>WEB</i>	9
3.	SEMANTIČKI <i>WEB</i> SERVISI.....	12
3.1.	SOAP	12
3.2.	WSDL	14
4.	NADZORNE KAMERE	17
4.1.	IP KAMERE	17
4.2.	PTZ KAMERE.....	17
5.	PRAKTIČNI RAD	18
5.1.	OPREMA	19
5.1.1.	IP PTZ kamera Sunell.....	20
5.1.2.	Router D-Link DIR-600	23
5.2.	RAČUNALNI SOFTVER	25
5.2.1.	WAMP.....	25
5.2.2.	PHP.....	25
5.2.3.	ONVIF	26
5.2.4.	SoapUI.....	26
5.2.5.	Ostali korišteni softver.....	27
5.3.	RAZVIJENA <i>WEB</i> APLIKACIJA	28
5.3.1.	HTML sučelje.....	28
5.3.2.	PHP skripte	31
5.4.	POTREBAN BROJ TOČAKA NADZORA	37
6.	ZAKLJUČAK	41

7.	LITERATURA	42
	Prilog 1: Izvorni kod test.html stranice	44
	Prilog 2: Izvorni kod test.php skripte	46
	Prilog 3: Izvorni kod test_preset.php skripte	50
	Prilog 4: Izvorni kod test_vlc.php skripte	52

POPIS SLIKA

Slika 1: Zaglavlje IPv4 paketa (prema [3])	3
Slika 2: ARPANET 1974. godine (prema [4])	4
Slika 3: ARPANET “postaje” Internet [5]	5
Slika 4: Internet postaje globalna mreža [5]	6
Slika 5: Penetracija fiksnog širokopojasnog interneta 2012. [5]	7
Slika 6: Penetracija mobilnog širokopojasnog interneta 2012. [5]	7
Slika 7: Prikaz jednostavne HTML <i>web</i> -stranice	9
Slika 8: Evolucija <i>weba</i> , odnosno semantike [10]	10
Slika 9: SOAP poruka	12
Slika 10: Fotografija prostora laboratorija s vidljivom opremom korištenom za rad ..	18
Slika 11: Fotografija prostora laboratorija sa strane kamere	18
Slika 12: Laptop, router i IP kamera korišteni za rad	19
Slika 13: Fotografija korištene IP kamere	20
Slika 14: Korisničko sučelje kamere s informacijama o uređaju	21
Slika 15: Konfiguracija IP adrese kamere	21
Slika 16: Specifikacije kamere	22
Slika 17: Drugi dio specifikacija kamere	22
Slika 18: Definiranje statičke IP adrese <i>routera</i>	23
Slika 19: <i>Virtual Server</i> podešavanja	24
Slika 20: Glavno sučelje <i>web</i> aplikacije	28
Slika 21: Izvršenje test_vlc.php skripte	35
Slika 22: Mapa za spremanje slika uhvaćenih VideoLAN reproduktorom	36
Slika 23: Snimka prostora bez optičkog zuma	38
Slika 24: Snimka prostora uz optički zum od 2,5x	39
Slika 25: Snimka jednog proizvoda uz optički zum od 11x	39
Slika 26: Prikaz deklaracije proizvoda uz zum od 20x	40

POPIS TABLICA

Tablica 1: Razvoj <i>weba</i> (prema [10]).....	10
Tablica 2: Kut pogleda u ovisnosti o razini optičkog zuma	37
Tablica 3: Horizontalni i vertikalni kutevi pogleda	37

SINKRONIZIRANO UPRAVLJANJE IP PTZ KAMEROM TEMELJENO NA SEMANTIČKIM MREŽNIM USLUGAMA

SAŽETAK

U ovom radu je opisan način rada semantičkih mrežnih servisa te njihova primjena. Dan je kratak pregled razvoja interneta do semantičkih mrežnih usluga. Praktični dio rada obavljen je na opremi koja uključuje IP PTZ nadzornu kameru Sunell s podrškom za komunikacijski protokol SOAP i ONVIF standard za kamere s podrškom za mrežni pristup. Izrađena je *web* aplikacija za upravljanje danom kamerom napisana u programskom jeziku PHP te je eksperimentalno utvrđen broj potrebnih točaka nadzora za nadzor prostora u ovisnosti o potrebama detaljnosti tog nadzora.

Ključne riječi: IP PTZ kamera, SOAP, ONVIF, mrežni servisi, semantički *web*

SYNCHRONIZED CONTROL OF AN IP PTZ CAMERA BASED ON SEMANTIC WEB SERVICES

ABSTRACT

This paper discusses functioning of the semantic web services and their application. The development of the Internet to semantic web services is described. The practical part of the paper was done on equipment including an IP PTZ surveillance camera which supports the SOAP communication protocol and ONVIF network camera standard. A camera control application was written in the PHP programming language and the number of surveillance points depending on the required level of surveillance detail was experimentally verified.

Keywords: IP PTZ camera, SOAP, ONVIF, web services, semantic web

1. UVOD

Rapidan razvoj informacijskih i komunikacijskih tehnologija od druge polovice 20. st. pa do danas omogućio je mnoge nove mogućnosti, kako u privatnoj, tako i u poslovnoj sferi. Svaki dan svjedočimo pojavi novih, sve moćnijih i sve bržih računala, komunikacijskih protokola i aplikacija, zadnjih godina također mobilnih uređaja koji su uvijek povezani na internet i koji cijeli sadržaj *weba* čine dostupnim na svakom koraku, a također sve više i raznih drugih uređaja koji se povezuju na internet u okviru *Internet of things* inicijative te je danas zaista moguće kupiti hladnjak, pećnicu ili klima uređaj s kojima se može komunicirati putem interneta te tako provjeravati liste za kupovinu namirnica, zadati temperaturu predgrijavanja pećnice ili željenu temperaturu stambenog prostora prije fizičkog dolaska u prostor; a *Smart TV-i*, odnosno televizori s internetskom vezom i podrškom za aplikacije, su postali praktički jedina vrsta televizora koja je dostupna u trgovinama.

Gdje je nekad bilo potrebno nekoliko dana, a u nekim slučajevima i nekoliko tjedana, kako bi obično pismo došlo od jedne do druge osobe, prijatelja ili poslovnih kolega ili partnera, korištenjem internetskih tehnologija, u prvom redu *e-maila* iliti elektroničke pošte, prijenos te vrste informacija je sveden na red veličine od jedne sekunde, pa i manje bez obzira na geografsku udaljenost između tih osoba. Kolege i poslovni partneri mogu istovremeno raditi na kompleksnim i povezanim zadacima, a primjerice strojarski tehnički crteži, CAD modeli se mogu crtati na jednom kraju svijeta i unutar nekoliko trenutaka, pogotovo zadnjih godina razvojem 3D pisača, biti fizički izrađeni na drugom.

Informacije, poput internetskih enciklopedija, stručnih i nestručnih tekstova, časopisa i knjiga, također su daleko dostupnije nego prije pojave interneta što uvelike smanjuje vrijeme potrebno da bi netko saznao bilo kakvu toj osobi bitnu informaciju.

Sve to dovodi do olakšavanja mnogih zadataka i povećanja kvalitete života, no istovremeno mijenja strukturu društva. Mnogi poslovi, pa čak i cijele industrije, su ugrožene zbog tih promjena. Elektronička pošta i servisi za trenutnu razmjenu poruka (*instant messaging*) uvelike umanjuju značaj klasične pošte, internetska televizija, radio i sajtovi namijenjeni reprodukciji medijskog sadržaja na zahtjev ugrozile su programsku televiziju i prodaju fizičkih glazbenih i filmskih medija, dok *web-shopping* dovodi do sve većeg broja zatvaranja, pogotovo malih, dućana...

Slično je i sa nadzorom, bilo sigurnosnim ili nadzorom rada strojeva, proizvodnih pogona, transporta ili drugog. Kamere koje koriste internetske protokole te im se može udaljeno pristupati i pratiti što one snimaju centraliziran nadzor mnogih poslovnih pothvata bez obzira na geografske lokacije bilo koje uključene strane.

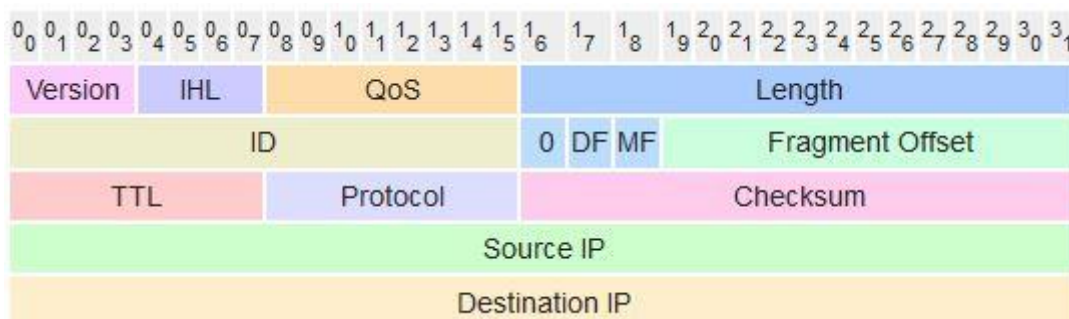
2. INTERNET

Internet se često naziva „mrežom mreža“, to je globalni sustav spojenih računalnih mreža koji se bazira na TCP/IP protokolu [1]. Sastoji se od mnogih privatnih, javnih, akademskih, poslovnih te drugih mreža, koje su povezane u cjelinu različitim tehnologijama žičanih i bežičnih veza.

2.1. POVIJEST INTERNETA

Razvoj interneta započinje 1960-ih godina u SAD-u na sveučilištu MIT (Massachusetts Institute of Technology) i u vladinoj agenciji DARPA (Defense Advanced Research Projects Agency) [2].

Omogućio ga je raniji razvoj tehnologije, odnosno metode paketnog prijenosa podataka (*packet switching*). Ta metoda grupira podatke koje se prenose u pakete koji se sastoje od zaglavlja koje sadrži informacije o pošiljatelju i primatelju paketa, duljini poruke i druge popratne informacije, te glavnog dijela, „tereta“ (*payload*), koji je ustvari poruka koju želimo prenijeti. Razmjena paketa povećanje efikasnosti i robusnosti mreže te istovremeno korištenje mreža od strane većeg broja uređaja [3]. Slika 1 prikazuje zaglavlje IPv4 paketa i što sve ono sadrži.



Slika 1: Zaglavlje IPv4 paketa (prema [3])

Radom MIT-a i DARPA-e tokom 1969. se u pogon stavlja ARPANET, koji se smatra prvom pravom pretečom današnjeg interneta, koji je uz paketni prijenos podataka koristio i TCP/IP koje je i danas osnova interneta. Tokom narednih godina se u ARPANET spaja sve više računala.

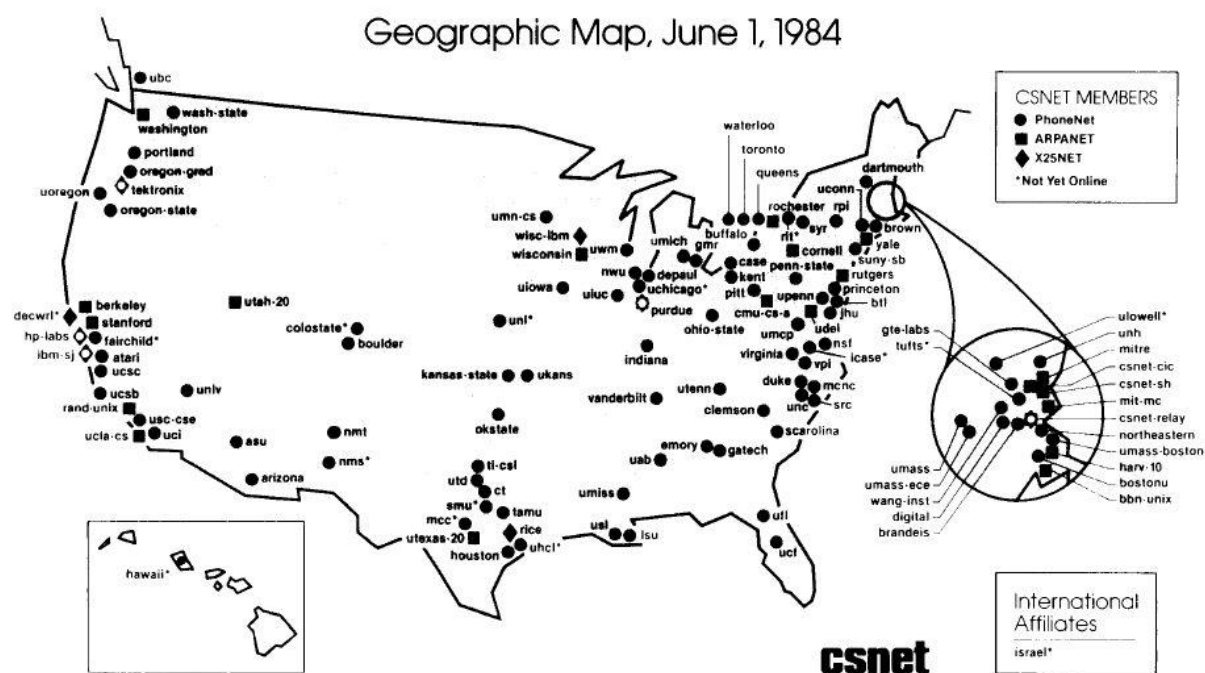


Slika 2: ARPANET 1974. godine (prema [4])

Na slici je vidljivo koliko je malen bio broj međusobno povezanih uređaja, u odnosu na današnji internet, te kako su prvi korisnici mreže bili vladine organizacije te edukacijske ustanove, odnosno sveučilišta.

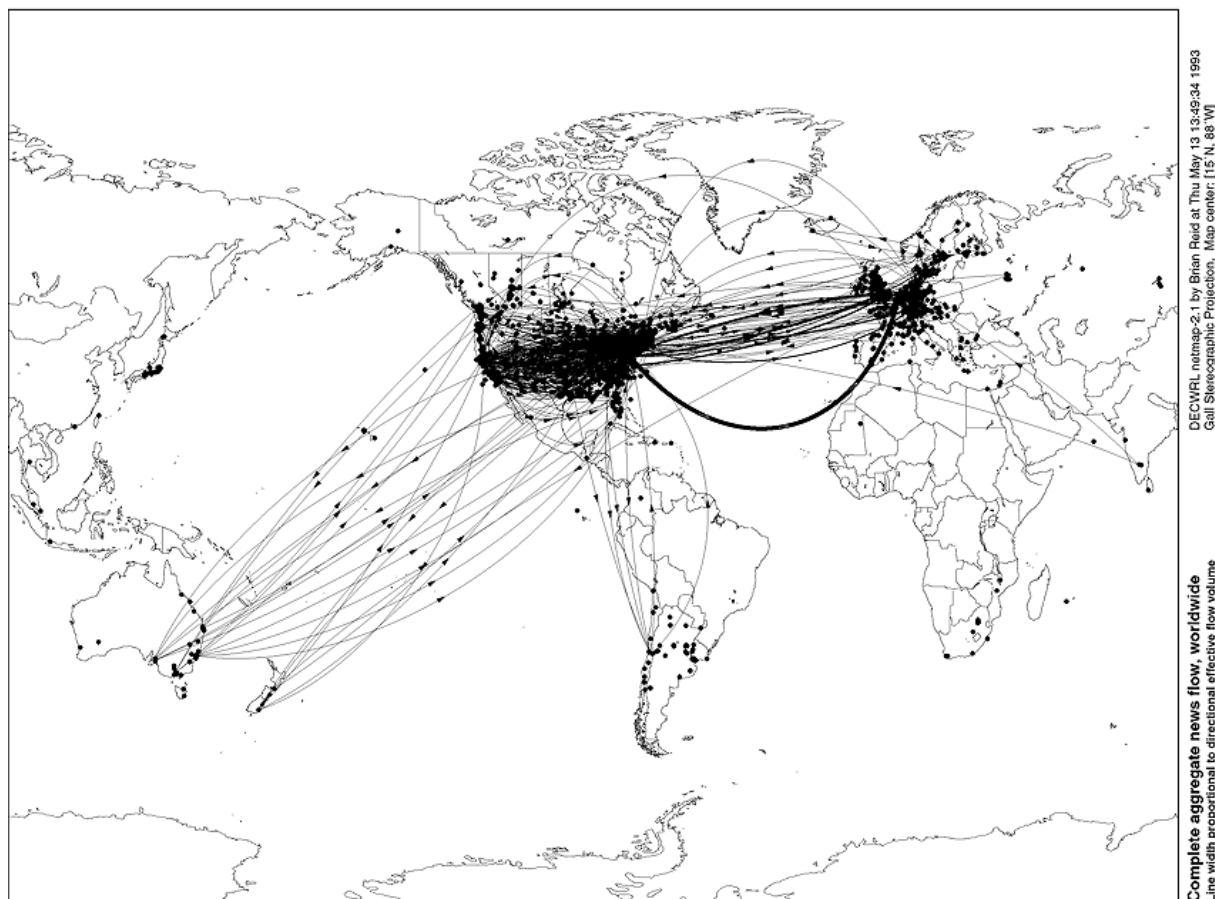
1972. godine je organizirana prva javna demonstracija sustava, a iste godine se razvija i elektronička pošta, odnosno te je godine napisan prvi softver za slanje elektroničkih poruka i poslana prva *e-mail* poruka te je ubrzo postala najbitnija mrežna aplikacija i to ostala naredno desetljeće, a vrlo je bitna i danas.

Internet je od početka zamišljan kao mreža mnogih međusobno spojenih mreža otvorene arhitekture, različitih sučelja za međusobno spajanje i u skladu s potrebama okruženja i korisnika pojedinih mreža bez strogih ograničenja. No praktičnost je ipak zahtijevala donošenje određenih koncepata i pravila kako bi komunikacija među različitim korisnicima bila moguća, poput algoritama za osiguranje od gubitka paketa u prijenosu i ponovnog slanja izgubljenih paketa, prolazak paketa kroz uređaje danas poznate kao *gateway* i *router* koji osiguravaju pravilno prosljeđivanje paketa te ne pamte podatke o njima radi jednostavnosti, potrebu za kontrolnim zbrojevima (*checksums*) koji osiguravaju ispravnost pristiglih paketa, za globalnim adresiranjem, komunikacijom između različitih operativnih sustava, te, također jedna od ključnih postavki koja vrijedi i danas, nepostojanje krovne organizacije za cijeli sustav koja bi globalno kontrolirala prijenos podataka.



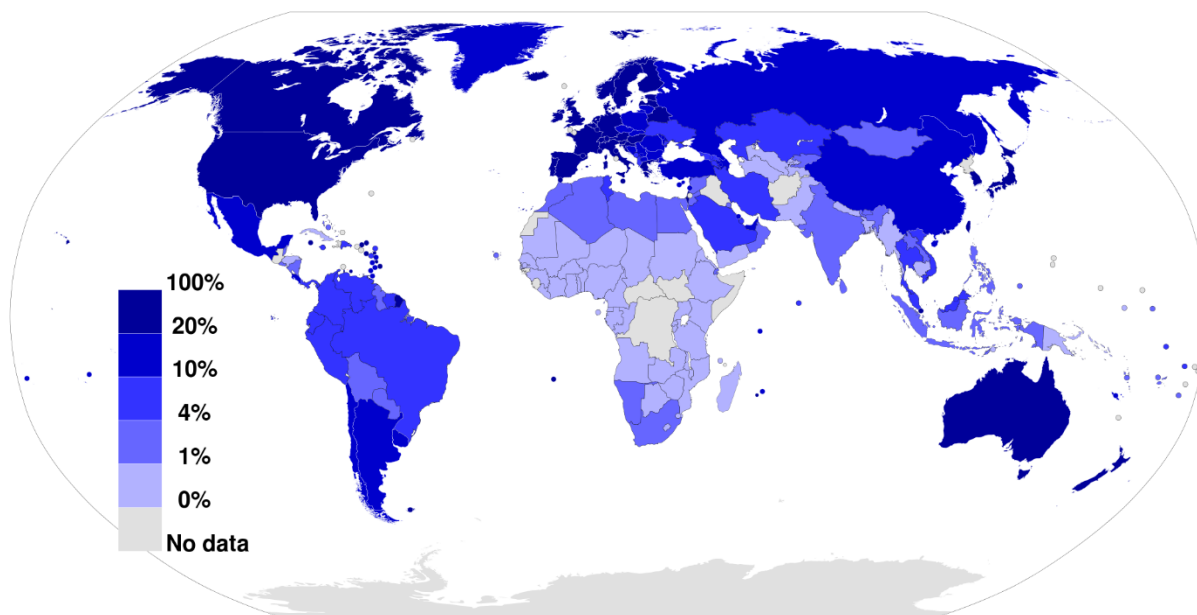
Slika 3: ARPANET "postaje" Internet [5]

Početkom 1980-ih započinje ubrzan rast mreža te se također, uz ARPANET, počinju pojavljivati druge, od kojih je jedna od prvih CSNET, čije spajanje s ARPANET-om se smatra pravim početkom interneta [5].

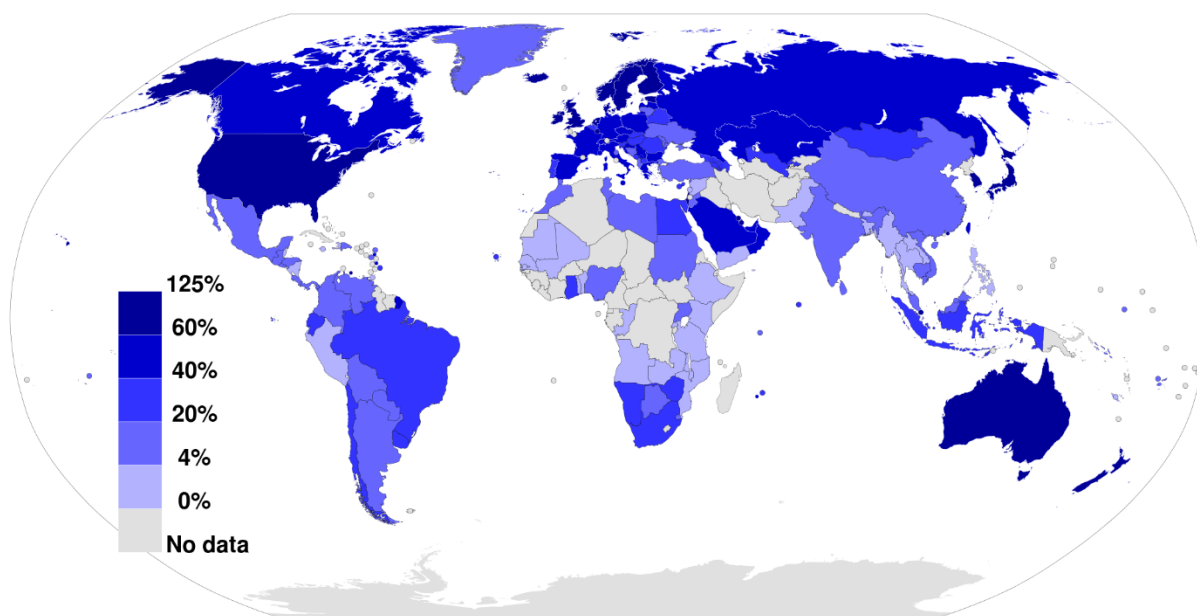


Slika 4: Internet postaje globalna mreža [5]

Slika 4 prikazuje globalnu upotrebu Useneta (aplikacije koja je omogućavala razmjenu poruka poput kasnijih foruma) 1993. godine. Vidljiv je vrlo rapidan razvoj broja korisnika interneta. Eksponencijalni razvoj se nastavlja do danas. Slika 5 i Slika 6 prikazuju koliko je korisnika širokopojsnih fiksnih i mobilnih internetskih veza u svijetu 2012. godine.



Slika 5: Penetracija fiksnog širokopojasnog interneta 2012. [5]



Slika 6: Penetracija mobilnog širokopojasnog interneta 2012. [5]

2.2. KLASIČNI WEB

Iako je *web* osmišljen kao informacijski prostor koji će biti koristan ne samo ljudima, nego i strojevima, dosad je većina informacija plasirana i korištena na internetu prvenstveno namijenjena korištenju ljudima te je malo informacija dovoljno dobro semantički i strukturno definirano da bi one zaista bile korisne i strojevima, odnosno računalima [6].

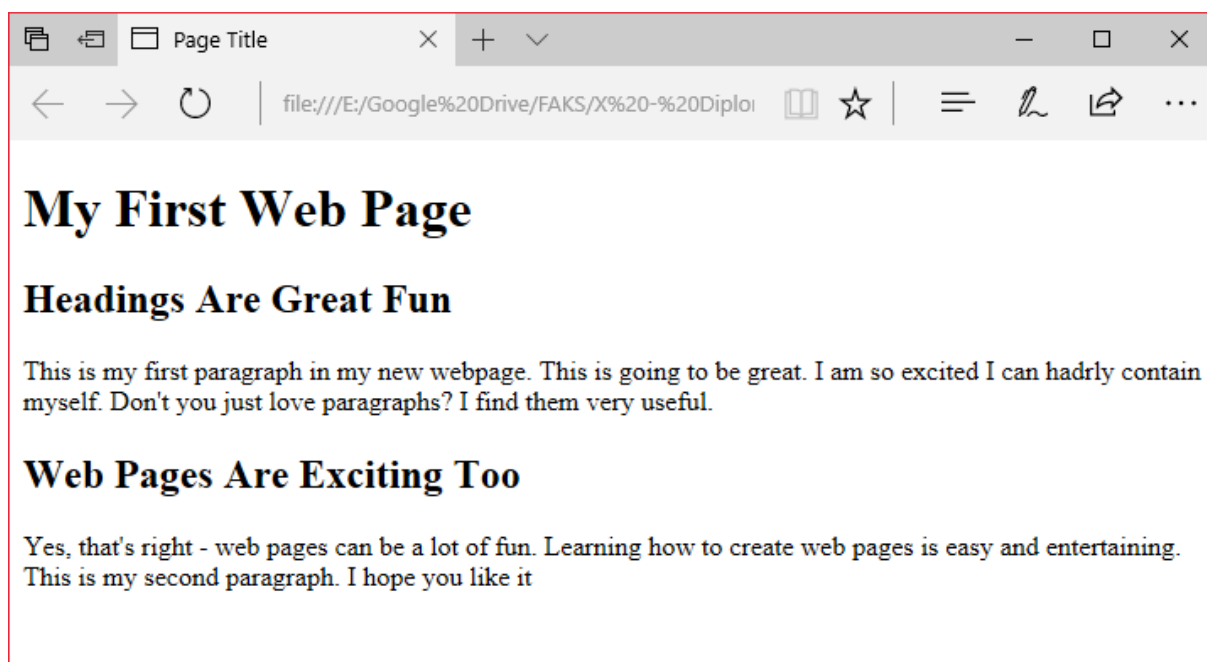
U skladu s tim, najveći dio današnjeg *weba* se sastoji od HTML dokumenata (*Hyper Text Markup Language*), što je jezik koristan za vizualnu prezentaciju te je njime lako ostvariti prikaz informacija za ljude, no računalna interpretacija tih dokumenata nije moguća.

Primjer HTML koda i rezultirajući prikaz u obliku stranice u internetskom pregledniku[7]:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Web Page</h1>
<h2>Headings Are Great Fun</h2>
<p>This is my first paragraph in my new webpage. This is going
to be great. I am so excited I can hardly contain myself.
Don't you just love paragraphs? I find them very useful. </p>

<h2>Web Pages Are Exciting Too</h2>
<p>Yes, that's right - web pages can be a lot of fun. Learning
how to create web pages is easy and entertaining. This is my
second paragraph. I hope you like it</p>
</body>
</html>
```



Slika 7: Prikaz jednostavne HTML web-stranice

2.3. SEMANTIČKI WEB

Semantički *web* upravo ima za cilj omogućiti realizaciju *weba* koji će moći napredovati i mijenjati početno znanje, pomoći ljudima da pronađu ono što znaju i odgovore na svoja pitanja te biti razumljiv i strojevima i ljudima [11]. Semantički *web* ima za cilj i omogućiti dinamičnost u smislu da sve informacije budu smislene i jednoznačno povezane sa drugim informacijama te tako stvoriti dublju i ekstenzivniju mrežu znanja, podataka i drugih vrsta informacija.

XML (*Extensible Markup Language*) je format za pohranu i prijenos podataka neovisan o softveru i hardveru [8]. Strukturu se XML fokusira na dokumente, ali ostvaruje široku primjenu za prikaz proizvoljnih podatkovnih struktura poput onih korištenih u *web* servisima.

Sličnost HTML-a i XML-a je u tome što su oba jezika jezici za označavanje. Unatoč tome, postoje velike razlike između njih, poglavito u njihovim ciljevima [8]. XML je stvoren s ciljem prijenosa podataka s fokusom na to što ti podaci predstavljaju, dok je primarna namjena HTML-a prikaz podataka s fokusom na izgled samih podataka. Druga je velika razlika što XML oznake nisu predefinisane, poput, u primjeru u potpoglavlju ranije, HTML oznaka tipa `<h1>` za poglavlja, `<p>` za paragrafe, već autor XML dokumenta ima slobodu izmisliti potpuno

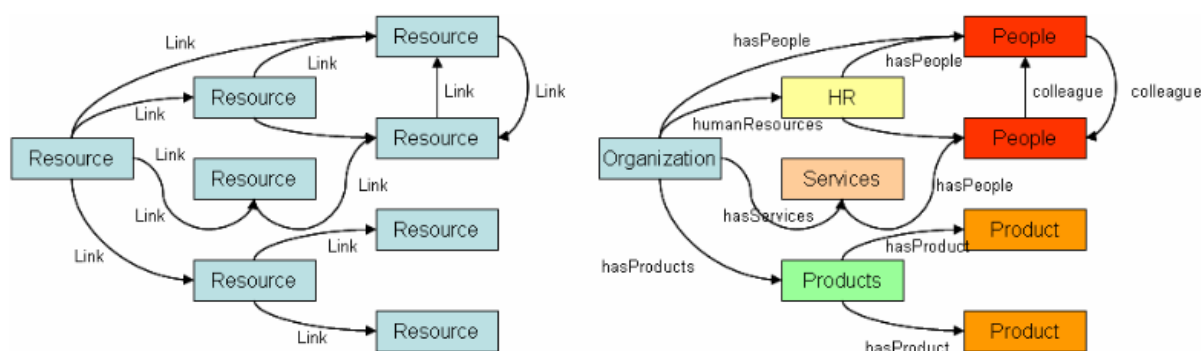
proizvoljne oznake te je jedino potrebno da primatelj dokumenta, odnosno aplikacija, zna pročitati te oznake.

Tablica 1 prikazuje razvoj *weba* od statičkog do semantičkog od 1990-ih godina do danas.

	Statički	Dinamički	Sintaksa	Semantika
Kodiranje	HTML	+RDBMS	+XML	+RDF/OWL
Izrada	Ručna	Generiranje serverskim aplikacijama	Generiranje aplikacijama bazirano na shemama	Generiranje aplikacijama bazirano na modelima
Korisnici	Ljudi	Ljudi	Ljudi i aplikacije	Ljudi i aplikacije
Paradigma	Prikaz	Stvaranje/Upiti/Ažuriranje	Integracija	Interoperabilnost
Aplikacije	Internetski preglednici	Internetski preglednici	Integracija procesa, EAI, BPMS, Tijekovi rada	Inteligentni agenti, Semantički <i>engine</i>

Tablica 1: Razvoj *weba* (prema [10])

Jedan od također ključnih pojmova vezanih za semantički *web* je RDF, odnosno *Resource Description Framework*, podatkovni model metapodataka, koji, za razliku od klasičnih oblika baza podataka koje koriste dvodimenzionalne tablice, koristi „trojke“, odnosno izraze u obliku subjekt-predikat-objekt [12]. Subjekt je ustvari resurs o kojem se radi, dok predikat objašnjava svojstva subjekta te njegovu vezu s objektom.



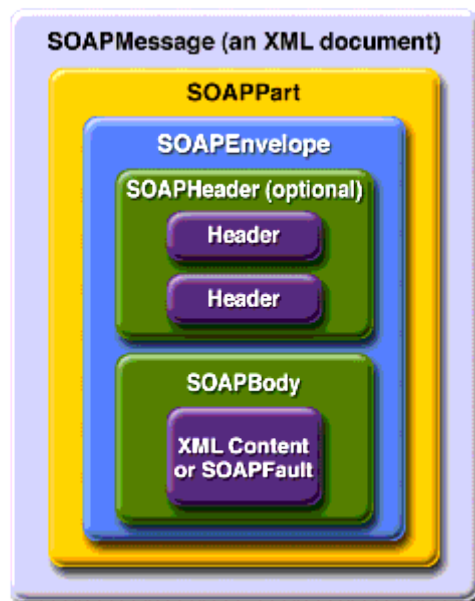
Slika 8: Evolucija *weba*, odnosno semantike [10]

Slika 8 prikazuje distinkciju između klasičnog *weba*, kod kojeg postoje samo veze između različitih informacija, dokumenata, te semantičkog kod kojeg se te informacije sklapaju u smislenu cjelinu.

3. SEMANTIČKI WEB SERVISI

3.1. SOAP

SOAP je otvoreni standard razmjene poruka HTTP protokolom između računala baziran na XML-u [13]. On omogućuje prijenos podataka za *web* servise i jednostavno slanje poruka, povezivanje s udaljenim servisima i pozivanje određenih programskih funkcija na udaljenim računalima ili drugim uređajima.



Slika 9: SOAP poruka

Slika 9 shematski prikazuje izgled SOAP poruke. Ona se sastoji od omotnice (*envelope*) u kojoj je sadržano zaglavlje i tijelo poruke. Omotnica je nužan element svake SOAP poruke koji definira početak i kraj poruke. Zaglavlje definira primatelja poruke, kao i dodatne informacije, poput korisničkog imena, šifre, korištenog sigurnosnog protokola i drugo. Tijelo poruke sadrži XML podatke o poruci te opcionalno dodatno informacije o mogućim greškama.

Sljedeći kod prikazuje strukturu općenite SOAP poruke, a u kasnijim će poglavljima biti prikazane specifične SOAP poruke korištene u okviru rada:

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://www.w3.org/2001/12/soap-envelope" SOAP-
ENV:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <SOAP-ENV:Header>
```

```
    ...  
    ...  
</SOAP-ENV:Header>  
  
<SOAP-ENV:Body>  
    ...  
    ...  
    <SOAP-ENV:Fault>  
        ...  
        ...  
    </SOAP-ENV:Fault>  
    ...  
</SOAP-ENV:Body>  
  
</SOAP_ENV:Envelope>
```

Ukoliko uređaj kojem je SOAP poruka poslana istu razumije tu poruku, on obavlja zahtjev iz poruke, npr. pretražuje bazu podataka za nekom informacijom, svoje stanje ili određene parametre, npr. vrijeme i datum, ili, u slučaju fizičkih i pomičnih uređaja, poput PTZ IP kamere, obavlja zahtjev za pomakom, kontinuiranom kretnjom ili zaustavljanjem.

3.2. WSDL

WSDL (*Web Service Definition Language*) se koristi za opis mrežnih servisa koje se mogu koristiti u porukama, odnosno operacijama između uređaja koje koriste te servise [15]. Pojednostavljeno, predstavlja opis metoda dostupnih za korištenje npr. u SOAP porukama. Također je baziran na XML formatu, a ovdje će biti predstavljen primjer WSDL dokumenta:

```
<?xml version="1.0"?>
<definitions name="StockQuote"

targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"
    xmlns:xsd1="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns="http://schemas.xmlsoap.org/wsdl/">

    <types>
        <schema
targetNamespace="http://example.com/stockquote.xsd"
            xmlns="http://www.w3.org/2000/10/XMLSchema">
            <element name="TradePriceRequest">
                <complexType>
                    <all>
                        <element name="tickerSymbol"
type="string"/>
                    </all>
                </complexType>
            </element>
            <element name="TradePrice">
                <complexType>
                    <all>
                        <element name="price" type="float"/>
                    </all>
                </complexType>
            </element>
```



```
</schema>
</types>

<message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
</message>

<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd1:TradePrice"/>
</message>

<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>

<binding name="StockQuoteSoapBinding"
type="tns:StockQuotePortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
        <soap:operation
soapAction="http://example.com/GetLastTradePrice"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
</binding>
```

```
<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort"
binding="tns:StockQuoteBinding">
    <soap:address
location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>
```

4. NADZORNE KAMERE

Kamere su moćan alat koji se koristi posljednjih desetljeća za zabilježavanje događaja. Njihova svrha se očituje u raznim aspektima: od zabavnih do istraživačkih. Međutim, razvojem memorije i načina prijenosa podataka, kamere su s vremenom dobile i druge zadaće – tako se pod pojmom nadzorne kamere najčešće smatraju kamere koje se koriste za zaštitu objekata, te se njihova snimka pohranjuje na određeni medij u nekom vremenskom periodu određenom memorijom medija i zahtjevima na duljinu trajanja snimke. No, razvojem softvera poboljšano je upravljanje kamerama te se danas nadzorne kamere često koriste za nadzor i analizu proizvodnih procesa, ali i, primjerice, praćenje stanja skladišta.

4.1. IP KAMERE

Pod pojmom IP kamera podrazumijevaju se kamere koje se mogu pratiti u realnom vremenu putem žične ili bežične mreže na udaljenom uređaju ili više njih. Također, kamerama je moguće manipulirati putem mreže, pritom određujući njihovu orijentaciju, razinu detalja i način pohrane podataka.

4.2. PTZ KAMERE

PTZ kamere koriste motorizirane dijelove da bi mogle ostvariti željenu orijentaciju u prostoru. PTZ je kratica za *Pan-Tilt-Zoom*, što osigurava 3 stupnja slobode gibanja, a navedeni pojmovi su objašnjeni u poglavlju o korištenoj kameri.

5. PRAKTIČNI RAD

Praktični dio ovog rada obavljen je na opremi Laboratorija za projektiranje izradbenih i montažnih sustava Fakulteta. Laboratorij je korišten kao prostor u kojem se obavlja nadzor IP kamerom.



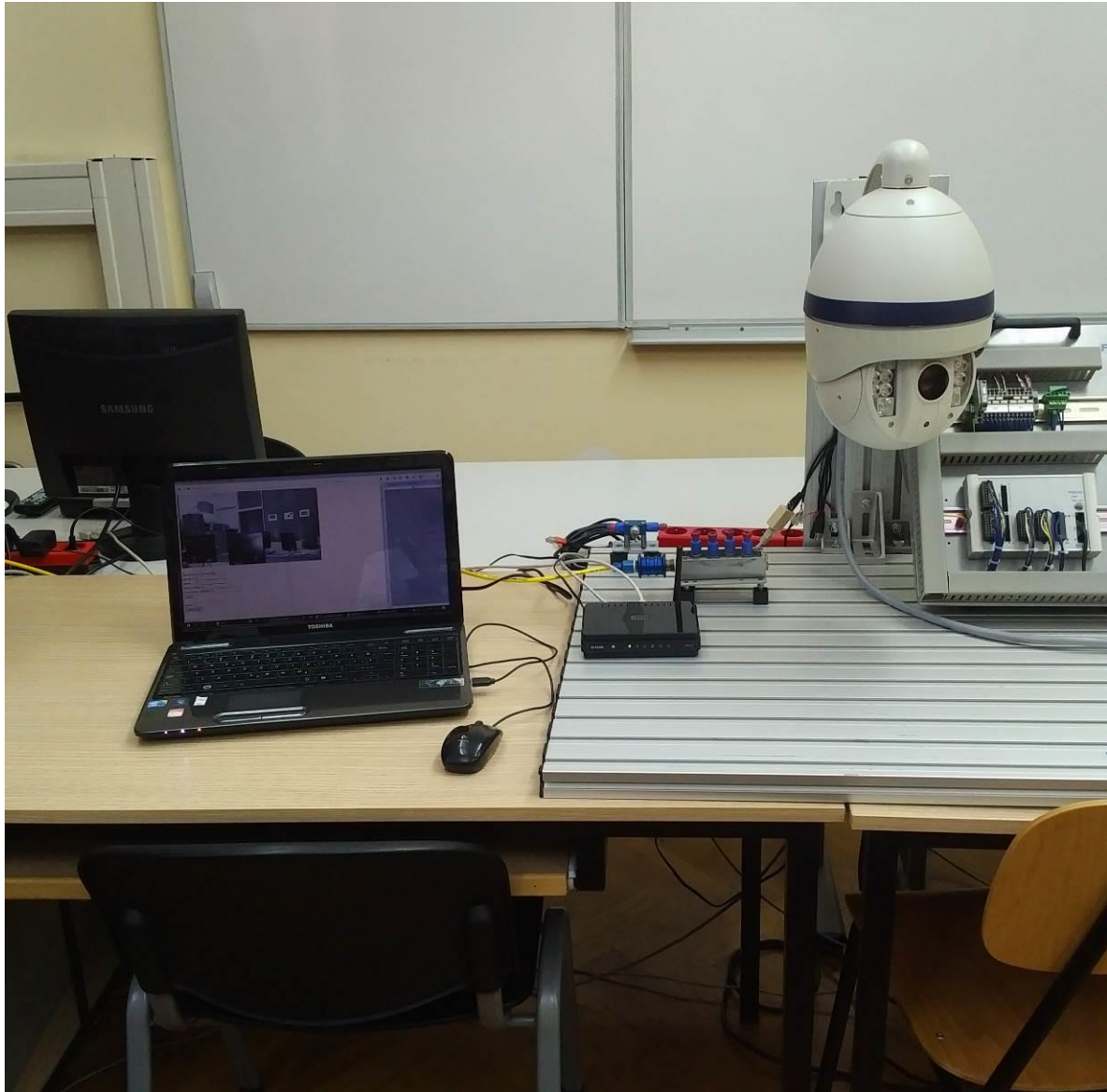
Slika 10: Fotografija prostora laboratorija s vidljivom opremom korištenom za rad



Slika 11: Fotografija prostora laboratorija sa strane kamere

5.1. OPREMA

Slika 12 prikazuje korištenu opremu unutar laboratorija. Korišten je Toshiba L655-1CK laptop s instaliranim Windows 10 OS-om. Bitniji dijelovi opreme bit će zasebno objašnjeni.



Slika 12: Laptop, router i IP kamera korišteni za rad

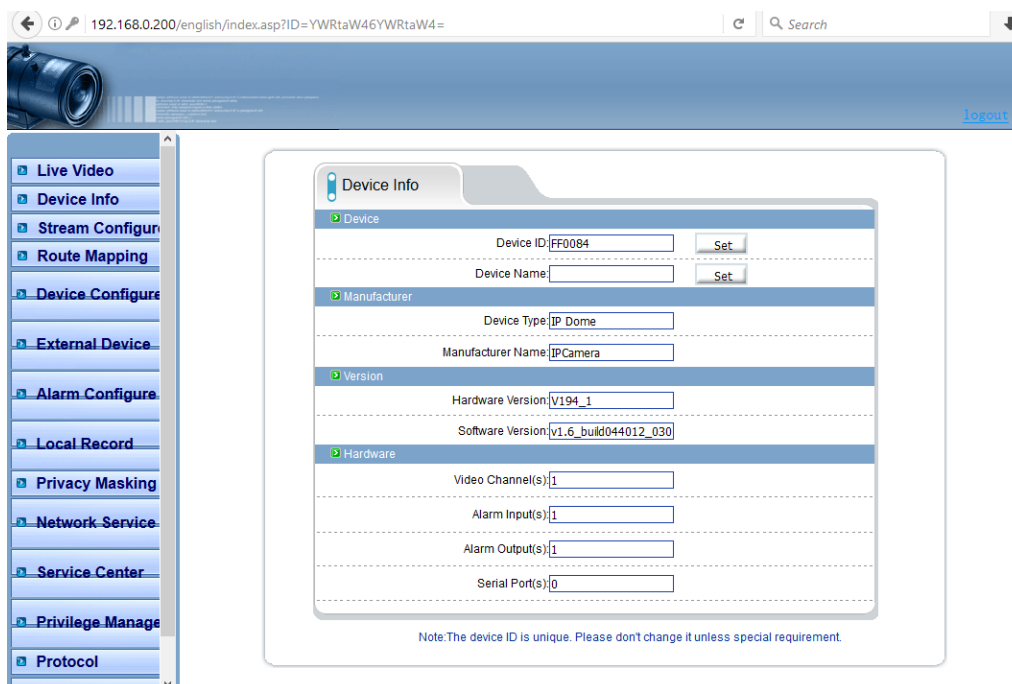
5.1.1. IP PTZ kamera Sunell

Pan-Tilt-Zoom kamera Sunell ima, kako joj ime kaže, mogućnosti *panninga*, odnosno okretanja oko jedne osi, za 360°, *tiltinga*, odnosno nagnjanja 90° te mogućnost zumiranja od 1 do 20x. Slika 13 prikazuje fotografiju kamere te je očit njen kupolasti oblik uvjetovan mogućnostima *panninga* i *tiltinga*. Rezolucija videa s kamere iznosi relativno niskih 704*384 piksela, no unatoč tome je uz pomoć visokog stupnja zuma moguće detaljno pregledavati prostor.



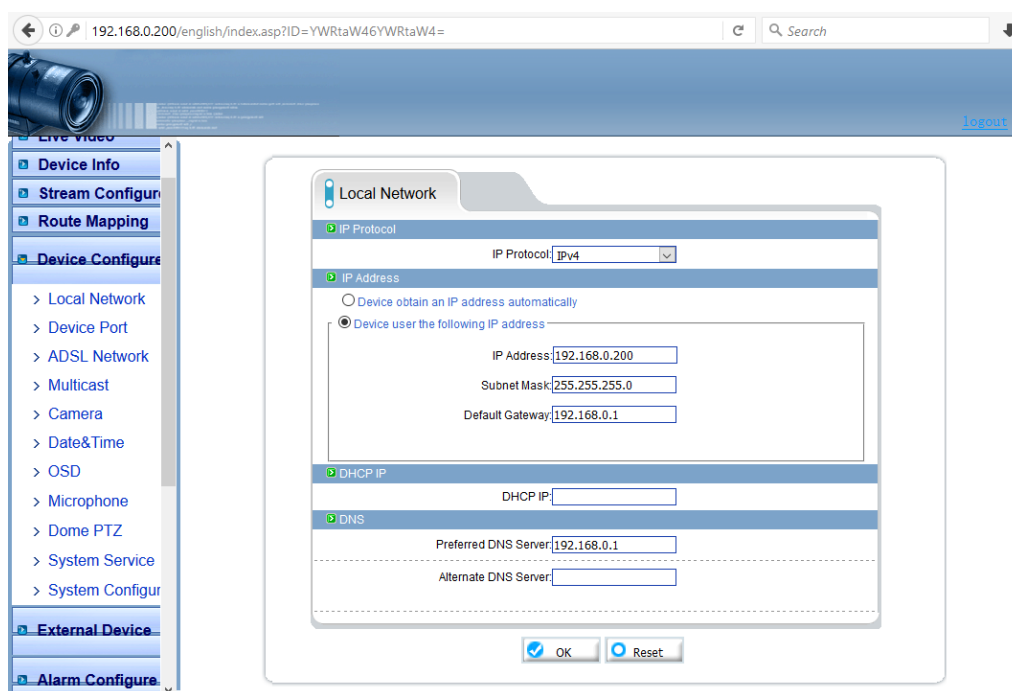
Slika 13: Fotografija korištene IP kamere

Slika 14 prikazuje informacije o uređaju dostupne u konfiguracijskom sučelju kod pristupa kameri preko lokalne mreže.



Slika 14: Korisničko sučelje kamere s informacijama o uređaju

Slika 15 prikazuje unos parametara potrebnih za dodjeljivanje statičke IP adrese kameri, koja je potrebna kako bi uređaju bilo moguće pristupati preko interneta, a ne samo koristeći lokalni mrežni priključak.



Slika 15: Konfiguracija IP adrese kamere

Slika 16 i Slika 17 prikazuju detaljne specifikacije kamere iz korisničkog priručnika.

Parameter	Description
Video out	1 BNC (PAL/NTSC, 1.0Vp-p, 75Ω)
Video Compression	H.264(ISO/IEC 14496-10) baseline profile to level 3.1
Image Frame Rate	PAL:D1(704x576)@25fps, CIF(352x288)@25fps, QCIF(176x144)@25fps; NTSC: D1(704x480)@30fps, CIF(352x240)@30fps, QCIF(240x160)@30fps.
Bit stream Control	CBR、VBR
Audio Input/Output	1 internal Microphone In, 1 External Microphone In, 1 Audio Line Out
Audio Compression	G723.1/6.3kbps
Audio bit stream	6.3Kbps
Web Server	Microsoft Internet Explorer Version 5.5 or higher
Network Protocol	RTP/RTCP,TCP/UDP,HTTP,DHCP, DNS,IPv4/IPv6,SMTP,FTP
Network Ethernet	RJ-45,10/100Base-T
Alarm Input/Output	1 Alarm Input
PTZ Control	RS485, Polco-D/Polco-P
SD Card	Support MicroSD/HC, MiniSD/HC
Safely	Watchdog, Password protection
Protection	IP66 Standard, Vandal Proof
CMS Client and SDK	Open API for application integration including SDK

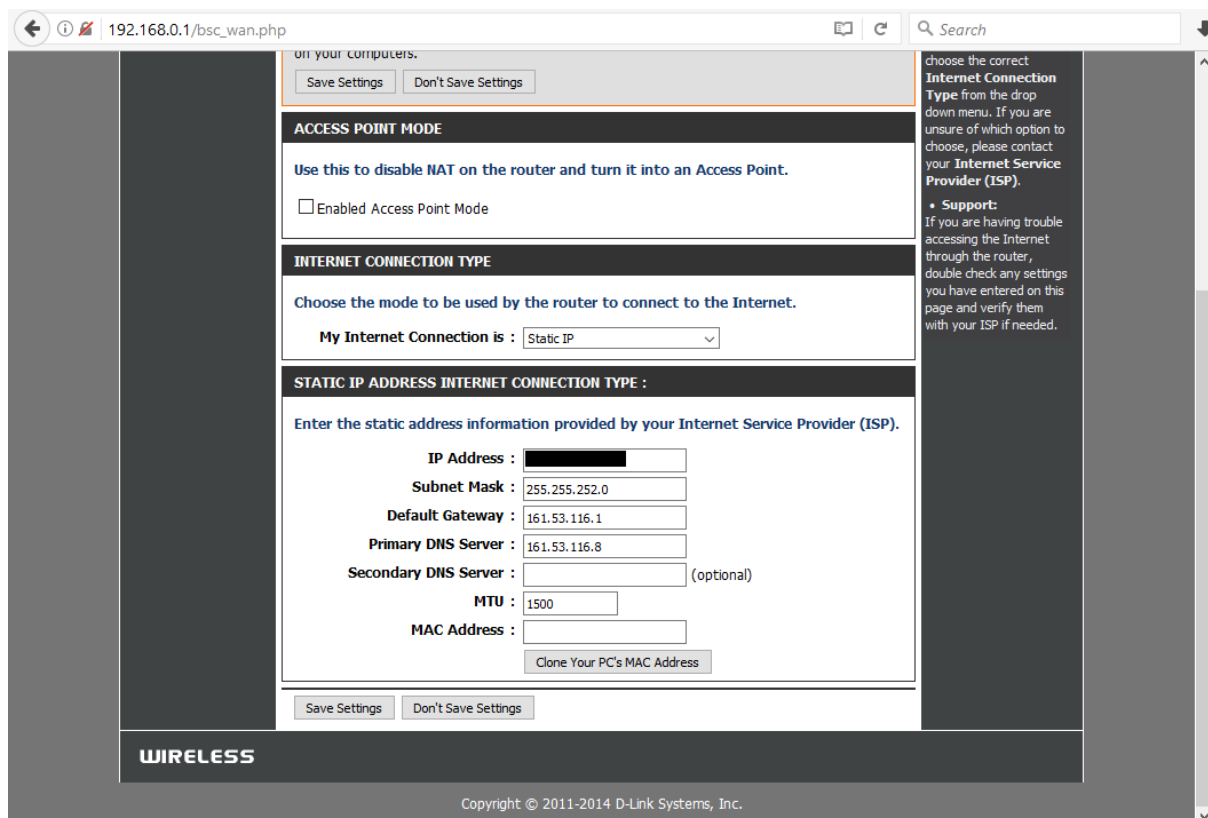
Slika 16: Specifikacije kamere

Electrical:		Set up:	
Rating Voltage	AC24V	Baud rate (RS485)	2400/4800/9600/19200bps
Power	12W indoor, 35W outdoor	Protocol	Pelco-P / Pelco-D/Phillips / Panasonic and so on, 17 Protocol choice
Operation:		Add. Set up	0-255
Decoder	Internal	Environment:	
Panning angle	360° Continuous rotation	Operation Environment	-40~ +60°C
Tilt angle	0°~90°	Humidity	0-95%No condensation
Rotary speed	Level 0.4~270°/S Vertical 0.4~160°/S	Protection level	IP66, 24-hour, safety cover TVS1500W lightning proof surge proof
Alarm	7 input, 2 output	Physical:	
Preset	220	Installation	Ceiling/Corner/ wall type
Surveillance way	Preset/touring/horizontal scan/ Auto learning	Weight(without bracket)	4.25Kg(outdoor)/3.85Kg(indoor)
Speed	Magnification times match with speed	Material	Aluminum

Slika 17: Drugi dio specifikacija kamere

5.1.2. Router D-Link DIR-600

Za rad je korišten *router* s mogućnošću žičnog i Wi-Fi bežičnog spajanja uređaja proizvođača D-Link[23]. Slika 18 prikazuje određivanje statičke IP adrese, potrebne za mogućnost direktnog pristupa kameri preko interneta, uz izbrisanu adresu iz sigurnosnih razloga.



Slika 18: Definiranje statičke IP adrese *routera*

Slika 19 prikazuje kako je podešena funkcija *Virtual Server routera*, koja omogućuje direktan pristup kameri prosljeđivanjem određenih portova adrese *routera* na portove i lokalnu IP adresu kamere



Slika 19: *Virtual Server* podešavanja

5.2. RAČUNALNI SOFTVER

5.2.1. WAMP

WAMP je skraćenica za *Windows-Apache-MySQL-PHP*. To je razvojno okruženje za Windows operativni sustav koje uključuje softverske pakete Apache2 kao poslužitelj *web*-stranica, podršku za programski jezik PHP, MySQL bazu podataka, te popratni upravljački, odnosno konfiguracijski softver. Omogućuje jednostavan početak razvoja *web* aplikacija bez potrebe za zasebnom instalacijom ili kompliciranom konfiguracijom pojedinačnih uključenih softverskih komponenti[16].

Apache, odnosno Apache HTTP Server je besplatni višepatformski, tj. dostupan za Linux, MacOS, FreeBSD te Windows operativne sustave, serverski softver otvorenog koda[17]. Serverski softver omogućuje računalu da postane *server* iliti poslužitelj, unutar njega se postavljaju *web*-stranice, HTML i bilo koje druge datoteke, izvršava aplikacije poput onih pisanih u PHP ili drugom programskom jeziku te se na njega pohranjuju baze podataka. Apache postoji od 1995. godine i ubrzo postaje jedan od najčešće korištenih softverskih paketa te vrsta, a danas je najpopularniji serverski softver te se koristeći njega poslužuje 46% svih aktivnih *web*-stranica.

MySQL je sustav za stvaranje relacijskih baza podataka (RDBMS, *relational database management system*) otvorenog koda. Ovaj rad nije zahtijevao korištenje njegovih mogućnosti stoga neće biti detaljnije opisivan.

5.2.2. PHP

PHP (rekurzivni akronim za *PHP: Hypertext Preprocessor*) je programski jezik za skriptiranje otvorenog koda posebno prilagođen za *web*-razvoj koji se može koristiti ugrađen u HTML stranice[18]. Za razliku od jezika koji se izvršavaju na klijentskom računalu poput JavaScripta, kod PHP-a se programski kod interpretira i izvršava na poslužitelju, kombinira s ostatkom sadržaja poput slika i drugog, poslužitelj generira HTML stranicu koja se tada šalje i prikazuje na klijentskom, odnosno računalu krajnjeg korisnika.

Strukturom uglavnom prati sintaksu programskog jezika C uz određene razlike koje su rezultat njegove primarne namjene za *web* razvoj. PHP varijable imaju prefix \$ radi bolje manipulacije *stringovima*, tj. varijablama koje sadrže tekst. Kod se može modularizirati upotrebom funkcija, odnosno metoda, a podržan je i stil pisanja orijentiran objektnom programiranju [20]. Kontrolne strukture uključuju *if*, *while*, *do/while*, *for*, *foreach* te *switch*. *If*

podrazumijeva uvjetno izvršavanje određene naredbe ukoliko je zadovoljen određeni uvjet, *while* petlje ponavljaju dio koda dok se ne postigne određeni uvjet ili dok je uvjet ispunjen, *for* petlje s brojačem služe višestrukom izvršavanju određenog dijela koda, a naredba *switch* ima sličnu funkcionalnost kao *if*, no uz veći broj mogućih vrijednost zadanog uvjeta.

U stranicama koje koriste PHP kod koriste se granične oznake `<?php` na početku dijela koji sadrži PHP kod i `?>` na kraju tog dijela. Svrha tih oznaka je odvajanje PHP koda od ostatka podataka sadržanih u datoteci, budući da se PHP uglavnom koristi unutar HTML datoteka te se time interpreteru koda govori koji dio datoteke treba izvršiti kao programski kod. Sav sadržaj izvan tih oznaka biva ignoriran od strane interpretera.

PHP cURL funkcija bit će korištena za slanje SOAP poruka u obliku HTML zahtjeva na kameru[19]

Osim kao dio HTML-a, PHP skripte mogu biti i samostalne datoteke, a moguće je i korištenje PHP-a u sučelju naredbenog retka.

5.2.3. ONVIF

Open Network Video Interface Forum je globalni otvoreni industrijski forum kojem je cilj izrada i unaprjeđivanje istoimenog standarda za sučelja fizičkih sigurnosnih proizvoda, u prvom redu nadzornih kamera, baziranih na internetskim tehnologijama[21]. Zahvaljujući ONVIF standardu moguća je interoperabilnost, odnosno mogućnost korištenja istih softverskih proizvoda za cijeli niz proizvoda različitih proizvođača.

Uključuje podršku za IP konfiguraciju proizvoda, otkrivanje kamera dostupnih na određenoj mreži, menadžment različitih uređaja, PTZ kontrolu, prikaz videa u realnom vremenu, programiranje alarma, primjerice kod otkrivanja subjekata u pokretu kao i automatsko praćenje subjekata te razne analitičke funkcije. Standard ne zahtijeva da svaki proizvod podržava sve funkcije već dostupne funkcije ovise o pojedinoj kameri.

5.2.4. SoapUI

SoapUI je alat otvorenog koda za testiranje SOAP i REST API-a (*Application Programming Interface*, odnosno sučelje za programiranje aplikacija)[22]. Njime je omogućeno učitavanje WSDL datoteka te testiranje SOAP funkcija sadržanih u tim datotekama.

U okviru ovog rada korišten je za testiranje te provjeru funkcija dostupnih na korištenoj IP kameri.

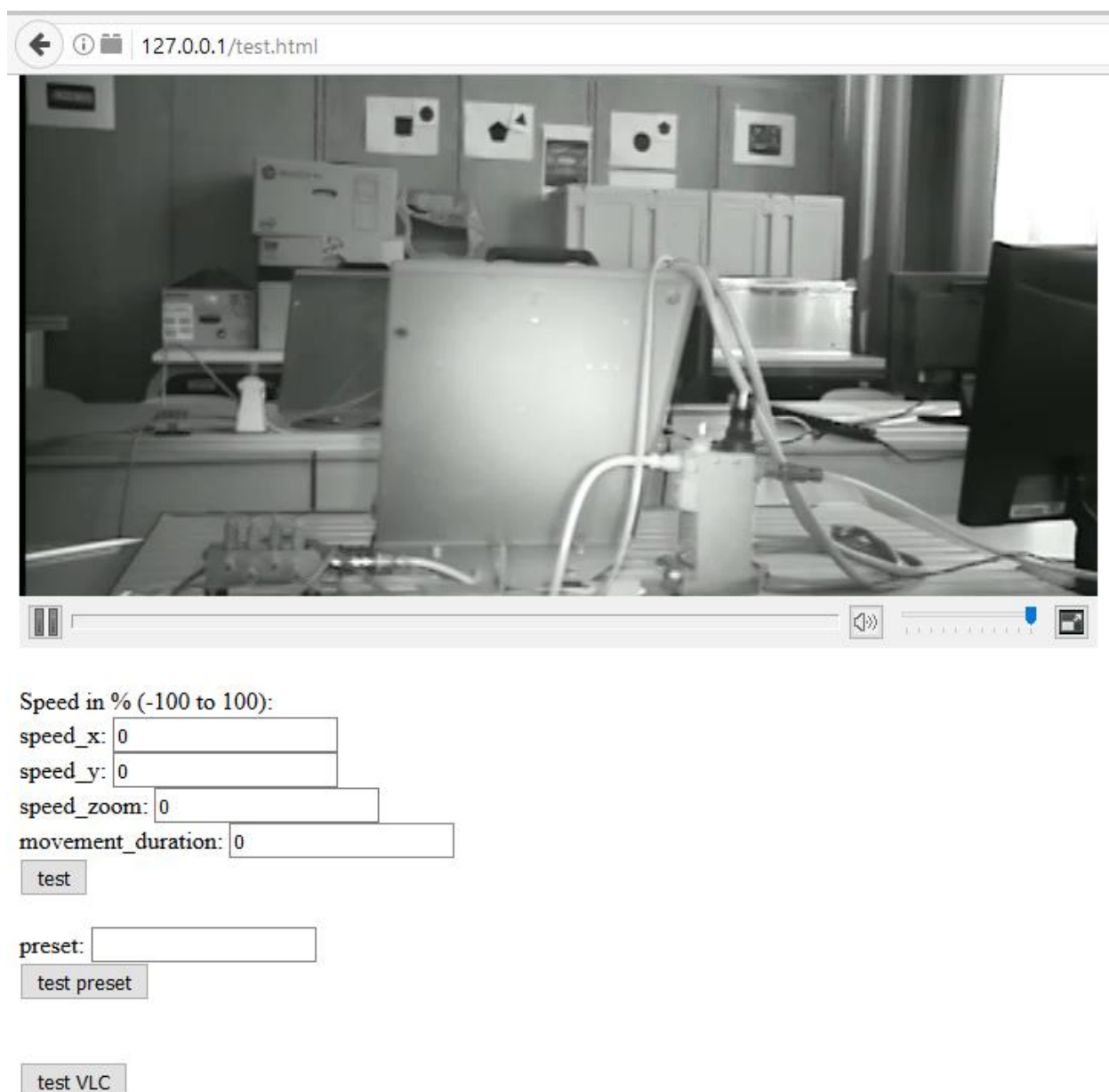
5.2.5. Ostali korišteni softver

Osim nabrojanih bitnijih softverskih paketa, kao internetski preglednik za testiranje razvijene *web* aplikacije korišten je Mozilla Firefox, VideoLAN medijski reproduktor korišten je kako bi se unutar aplikacije mogao dobiti prikaz slike sa kamere uživo, a za obradu programskog koda korišten je alat Notepad++.

5.3. RAZVIJENA WEB APLIKACIJA

5.3.1. HTML sučelje

Aplikacija napisana u sklopu ovog rada sastoji se od HTML stranice, kao glavnog sučelja, te PHP skripti koje ostvaruju funkcije aplikacije. HTML stranica sadrži prikaz videa sa kamere u realnom vremenu, koji je ostvaren koristeći VideoLAN medijski reproduktor unutar tzv. cctv kontejnera, te dvije HTML forme, od kojih se u jednu upisuju željene vrijednosti za ručno pomicanje kamere, a druga se koristi za pomicanje kamere do određene *preset* pozicije. Slika 20 prikazuje sučelje aplikacije.



Slika 20: Glavno sučelje web aplikacije

Cjelovit HTML kod nalazi se u prilogu, a ovdje će biti pojedinačno objašnjene relevantne stavke.

Sljedeći dio koda omogućuje prikaz snimke kamere u realnom vremenu koristeći VLC plugin za Firefox internetski preglednik koristeći RTSP *streaming* protokol:

```
<div id="cctv-container">
  <object classid="clsid:9BE31822-FDAD-461B-AD51-BE1D1C159921"
  codebase="http://downloads.videolan.org/pub/videolan/vlc/lates
  t/win32/axvlc.cab" id="vlc" events="True" >
    <param name="Src"
    value="rtsp://161.53.117.178:554/snl/live/1/1/Ux/sido=-
    Ux/sido="/>
    <param name="ShowDisplay" value="True"/>
    <param name="AutoLoop" value="False"/>
    <param name="AutoPlay" value="True"/>
    <embed id="vlcEmb" type="application/x-google-vlc-plugin"
    version="VideoLAN.VLCPlugin.2" autoplay="yes" loop="no"
    target="rtsp://161.53.117.178:554/snl/live/1/1/Ux/sido=-
    Ux/sido=" width="704" height="384"/>
  </object>
</div>
```

Forma u narednom dijelu koda omogućuju unos vrijednosti brzine gibanja kamere po tri moguće osi, te željeno trajanje tog gibanja, dok se pritiskom na tipku „test“ vrijednosti iz te forme metodom POST šalju skripti test.php koja obavlja željeno gibanje:

```
Speed in % (-100 to 100):
<form action="test.php" method="post">
speed_x: <input type="text" name="speed_x" value = 0
onfocus="if (this.value==0) this.value=' ' onblur="if
(this.value==' ') this.value=0"><br>
speed_y: <input type="text" name="speed_y" value = 0
onfocus="if (this.value==0) this.value=' ' onblur="if
(this.value==' ') this.value=0"><br>
```

```
speed_zoom: <input type="float" name="speed_zoom" value = 0
onfocus="if (this.value==0) this.value=''" onblur="if
(this.value=='') this.value=0"><br>
movement_duration: <input type="float"
name="movement_duration" value = 0 onfocus="if (this.value==0)
this.value=''" onblur="if (this.value=='') this.value=0"><br>
<input type="submit" name="test" value="test">
</form>
```

Forma koja omogućuje pokretanje kamere do pozicije određenog *preseta* izgleda ovako te omogućuje unos broja željene *preset* pozicije, a pritiskom gumba „test preset“ vrijednosti se šalju skripti `test_preset.php` koja ostvaruje to gibanje slanjem SOAP zahtjeva kameri:

```
<form action="test_preset.php" method="post">
preset: <input type="text" name="preset"><br>
<input type="submit" name="test_preset" value="test preset">
</form>
```

Posljednja forma služi pokretanju `test_vlc.php` skripte koja će biti objašnjena u sljedećem dijelu:

```
<form action="test_vlc.php" method="post">
<input type="submit" name="test_vlct" value="test VLC">
</form>
```


5.3.2. PHP skripte

Skripta test.php na početku sadrži inicijalizaciju cURL varijable `$ch` koja će biti korištena za ostvarivanje komunikacije s IP kamerom te zaglavlje HTTP zahtjeva koji se njom šalje, a koji sadrži IP adresu kamere, verziju protokola te definiciju vrste sadržaja zahtjeva kao SOAP+XML te druge podatke:

```
$ch = curl_init();
$headers = [
    'POST http://161.53.117.178:10000/ HTTP/1.1',
    'Accept-Encoding: gzip,deflate',
    'Content-Type: application/soap+xml; charset=UTF-8; action="http://www.onvif.org/ver10/device/wsdl/GetSystemDateAndTime"',
    'Connection: Keep-Alive',
];
```

Varijable brzine i trajanja u sljedećem dijelu koda bit će korištene u samom SOAP zahtjevu, a dobivene su POST metodom preko glavnog HTML sučelja objašnjenog u prethodnom potpoglavlju:

```
$speed_x = $_POST["speed_x"];
$speed_x = $speed_x / 100;
$speed_y = $_POST["speed_y"];
$speed_y = $speed_y / 100;
$speed_zoom = $_POST["speed_zoom"];
$speed_zoom = $speed_zoom / 100;
$movement_duration = $_POST["movement_duration"];
$movement_duration = $movement_duration * 1000000;
```

Brzine se dijele sa 100 kako bi se u glavno sučelje mogli upisivati cjelobrojni postoci umjesto decimalnih brojeva, dok se trajanje množi sa 1000000 kako bi se dobila potrebna brojka u milisekundama.

Naredni dio koda sadrži stvarni SOAP zahtjev koji se šalje kameri. Sadrži varijable brzine dobivene u prošlom dijelu te se može vidjeti struktura SOAP zahtjeva, sa zaglavljem koje sadrži informacije poput korištene WSDL datoteke, sigurnosnih informacija, odnosno

korisničkog imena i šifre te drugih detalja. Naredba `curl_exec($ch)`; šalje taj zahtjev kameri:

```
curl_setopt($ch, CURLOPT_POSTFIELDS,
'<soap:Envelope xmlns:sch="http://www.onvif.org/ver10/schema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wSDL="http://www.onvif.org/ver20/ptz/wSDL">
  <soap:Header><wsse:Security soap:mustUnderstand="true"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"><wsse:UsernameToken
wsu:Id="UsernameToken-
BBBE5AE173D8C1D5E41493907222924"><wsse:Username>admin</wsse:U
sername><wsse:Password Type="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">admin</wsse:Password><wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-
1.0#Base64Binary">/gkrSsnK+mMs2yFByrTr1Q==</wsse:Nonce><wsu:Cr
eated>2017-05-
04T14:13:42.292Z</wsu:Created></wsse:UsernameToken></wsse:Secu
rity></soap:Header>
  <soap:Body>
    <wSDL:ContinuousMove>

<wSDL:ProfileToken>profile_cam1_stream1</wSDL:ProfileToken>
    <wSDL:Velocity>
      <!--Optional:-->
      <sch:PanTilt x=" " . $speed_x . " y=" " . $speed_y
. " />

      <!--Optional:-->
      <sch:Zoom x=" " . $speed_zoom . " />
    </wSDL:Velocity>
```

```

        '//<wsdl:Timeout>' . $movement_duration .
'</wsdl:Timeout>
. '      </wsdl:ContinuousMove>
      </soap:Body>
</soap:Envelope>');
curl_exec($ch);

```

Nakon toga se skripta pauzira na vrijeme određeno varijablom `$movement_duration` naredbom `usleep($movement_duration);`

Šalje se još jedan SOAP zahtjev, onaj za zaustavljanjem kamere (prikazan skraćeno radi jednostavnosti), te se otpušta varijabla `$ch`:

```

curl_setopt($ch, CURLOPT_POSTFIELDS,
'<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-
(...)'
soap:Header>
  <soap:Body>
    <wsdl:Stop>

<wsdl:ProfileToken>profile_cam1_stream1</wsdl:ProfileToken>
    <!--Optional:-->
    <wsdl:PanTilt>0</wsdl:PanTilt>
    <!--Optional:-->
    <wsdl:Zoom>0</wsdl:Zoom>
  </wsdl:Stop>
</soap:Body>
</soap:Envelope>');
curl_exec($ch);
curl_close ($ch);

```

Kraj skripte vraća internetski preglednik na prethodnu stranicu, odnosno glavno sučelje `test.html`:

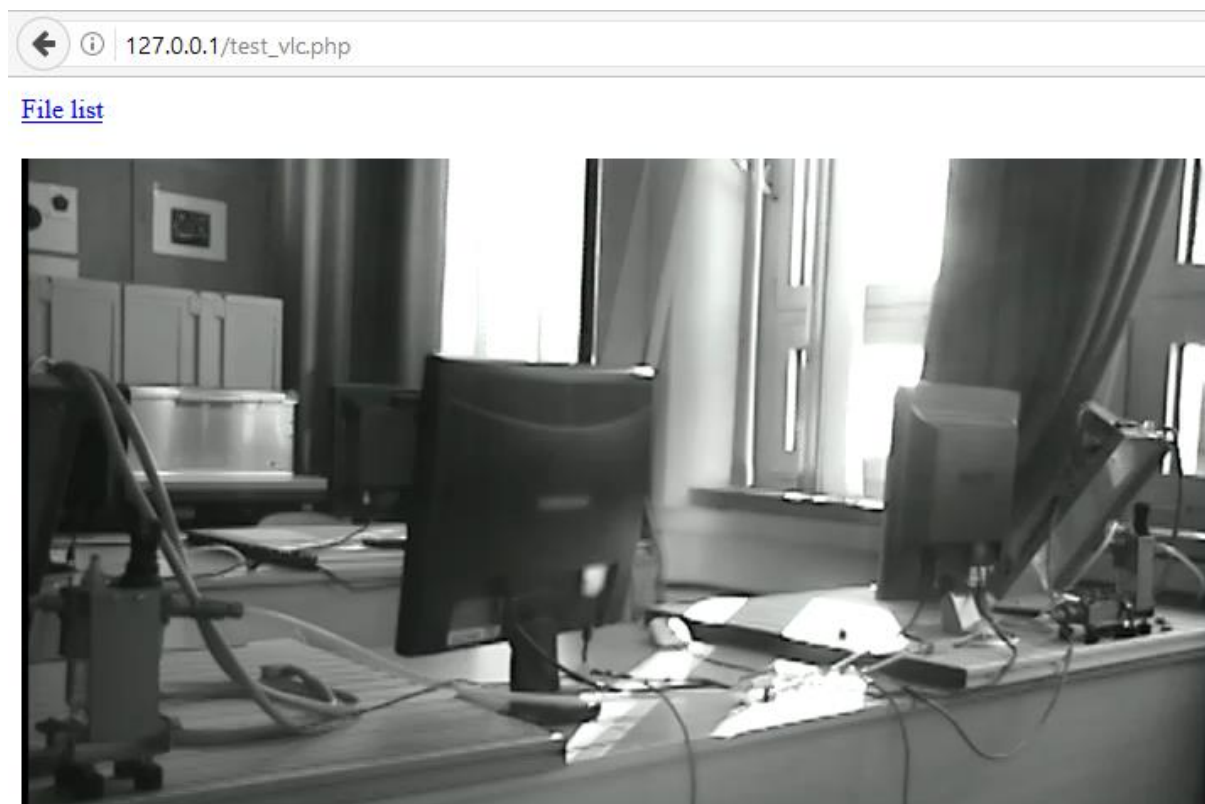
```
header("Refresh: 0, url=test.html");
```

Skripta `test_preset.php` je jednostavnija te se u njoj samo šalje SOAP zahtjev definiran u ONVIF standardu, odnosno WSDL datoteci kamere, koji šalje naredbu kameri da se maksimalnom brzinom pomakne na određenu *preset* poziciju. Bit će prikazan samo SOAP zahtjev u kojem je vidljiva naredba `wsdl:GotoPreset`:

```
'<soap:Envelope xmlns:sch="http://www.onvif.org/ver10/schema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsdl="http://www.onvif.org/ver20/ptz/wsdl">
  <soap:Header><wsse:Security soap:mustUnderstand="true"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"><wsse:UsernameToken
wsu:Id="UsernameToken-
BBBE5AE173D8C1D5E414939093288358"><wsse:Username>admin</wsse:U
sername><wsse:Password Type="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">admin</wsse:Password><wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-
1.0#Base64Binary">/eg7j/IzeCV3SRsgkdoENQ==</wsse:Nonce><wsu:Cr
eated>2017-05-
04T14:48:48.835Z</wsu:Created></wsse:UsernameToken></wsse:Secu
rity></soap:Header>
  <soap:Body>
    <wsdl:GotoPreset>

<wsdl:ProfileToken>profile_cam1_stream1</wsdl:ProfileToken>
    <wsdl:PresetToken>preset' . $_POST["preset"] .
'</wsdl:PresetToken>
    </wsdl:GotoPreset>
  </soap:Body>
</soap:Envelope>');
```

Skripta `test_vlc.php` u pozadini poziva VideoLAN medijski reproduktor, dakle bez pozivanja njegovog grafičkog sučelja, VideoLAN učitava RTSP *videostream* s kamere, te dohvaća nekoliko slika.



Slika 21: Izvršenje `test_vlc.php` skripte

Slika 21 prikazuje izgled rezultirajuće stranice. Prikazana je sama slika te poveznica na popis uhvaćenih slika. Slijedi kod glavnog dijela skripte:

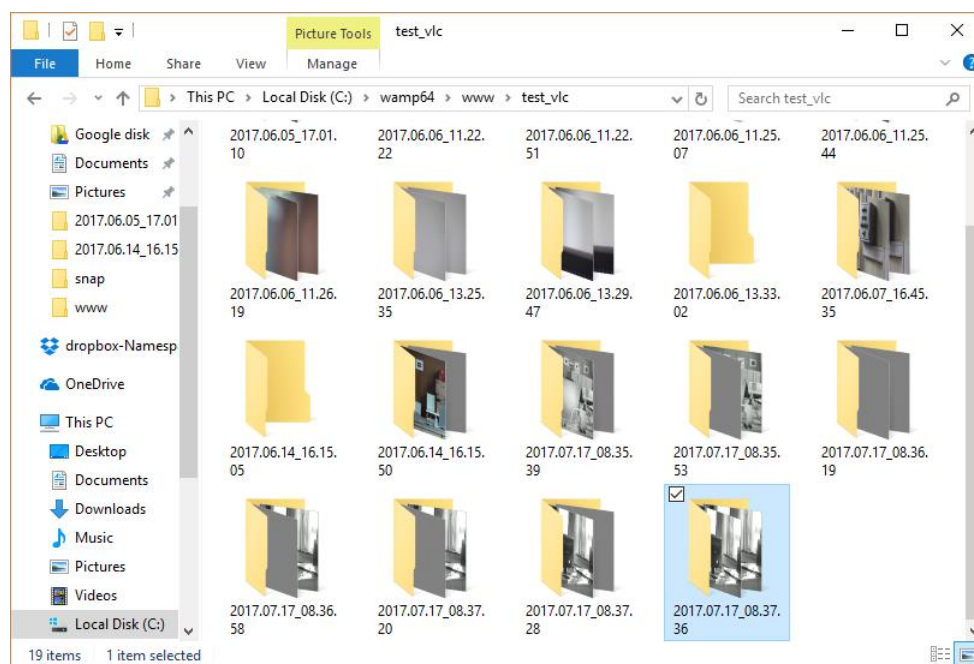
```
$time = date('Y.m.d_H.i.s');  
//$vlc = 'C:\Progra~2\VideoLAN\VLC\vlc.exe  
rtsp://192.168.0.200:554/snl/live/1/1/Ux/sido=-Ux/sido= --  
rate=1 --video-filter=scene --vout=dummy --scene-format=png --  
scene-ratio=12 --scene-prefix=snap --scene-  
path=C:\wamp64\www\test_vlc\' . $time . '\\ --run-time=2  
vlc://quit';  
$vlc = 'C:\Progra~2\VideoLAN\VLC\vlc.exe  
rtsp://161.53.117.178:554/snl/live/1/1/Ux/sido=-Ux/sido= --  
rate=1 --video-filter=scene --vout=dummy --scene-format=png --  
scene-ratio=12 --scene-prefix=snap --scene-
```

```

path=C:\wamp64\www\test_vlc\\" . $time . '\\ --run-time=2
vlc://quit';
echo("<a href=\""/test_vlc/" . $time . "\">File list</a>");
echo('<br><br>');
exec('mkdir C:\wamp64\www\test_vlc\\" . $time);
exec($vlc);
//header("Refresh: 0, url=test.html");
echo("<img src=\""/test_vlc/" . $time . "/snap00001.png\"/>");
echo (new TesseractOCR("/test_vlc" . $time .
"/snap00001.png"))
    ->psm(11)
    ->run();
echo('<br><br>');

```

Varijabla `$time` definira vremensku varijablu koja se koristi za određivanje mape u koju će VideoLAN spremati slike. Funkcija `exec('mkdir C:\wamp64\www\test_vlc\\" . $time);` kreira mapu sa zapisom vremena, `exec($vlc);` pokreće VideoLAN sa svim parametrima definiranim u ranijem dijelu koda, a `echo("");` daje prikaz te slike unutar stranice. Slika 22 prikazuje mapu s dobivenim slikama.



Slika 22: Mapa za spremanje slika uhvaćenih VideoLAN reproduktorom

5.4. POTREBAN BROJ TOČAKA NADZORA

Kako bi se odredio prostor kojeg kamera može uhvatiti, potrebno je odrediti minimalan broj točaka u koje je potrebno usmjeriti kameru da bi se osigurala dovoljna razina detaljnosti kojom se može pregledati prostor. Za određivanje potrebnog broja točaka koristio se eksperimentalni pristup, pošto specifikacije kamere nisu dovoljno detaljne da bi se iz njih mogli iščitati potrebni podaci.

Korištena IP PTZ kamera posjeduje karakteristiku optičkog zuma do 20x, koji određuje broj potrebnih točaka u prostoru. Naime, ukoliko se ne koristi zum, broj točaka je minimalan zbog kuta pogleda koji je u tome slučaju najveći u smjeru obje osi. Nasuprot tome, broj točaka je maksimalan kada je zum podešen na najveću razinu, koja iznosi već spomenutih 20x.

U tablici 2 prikazane su izračunate vrijednosti kutova pogleda potrebnih za definiranje broja točaka.

Optički zum	Horizontalni kut pogleda
1x	51,5°
2.5x	20,69°
11	4,78°
20	2,39°

Tablica 2: Kut pogleda u ovisnosti o razini optičkog zuma

Nakon izračuna horizontalnih kutova pogleda, potrebno je izračunati i vertikalni kut pogleda koji se izračunava kao:

$$v \text{ kut pogleda} = h \text{ kut pogleda} * \frac{\text{rezolucija u v smjeru}}{\text{rezolucija u h smjeru}},$$

a čiji je pregled dan u tablici 3.

Horizontalni kut pogleda	Vertikalni kut pogleda
51,5°	27,93°
20,69°	11,28°
4,78°	2,61°
2,39°	1,30°

Tablica 3: Horizontalni i vertikalni kutevi pogleda

Nakon izračuna horizontalnih i vertikalnih kutova pogleda, moguće je odrediti broj točaka u prostoru koje je moguće pokriti kamerom na sljedeći način:

- očitati horizontalnu pokrivenost prostora kamerom

- odrediti vertikalnu pokrivenost prostora kamerom
- izračunati broj potrebnih točaka pomoću već određenih kutova pogleda.

Horizontalna pokrivenost prostora kamerom iznosi 360° , a iščitana je iz specifikacija kamere, dok je vertikalna pokrivenost prostora jednaka zbroju ispruženog kuta i već izračunatog vertikalnog kuta pogleda. Izračunati broj potrebnih točaka proizlazi iz omjera pokrivenosti i kuta pogleda, a prikazan je u sljedećim primjerima, ilustriranih slikama 23, 24, 25 i 26.



Slika 23: Snimka prostora bez optičkog zuma

Na slici 23 je prikazan pogled kamerom bez aktivacije optičkog zuma, odnosno uz vrijednost optičkog zuma 1x. Vidljiva je velika kutna pokrivenost prostora, ali uz ograničenu razinu detalja.

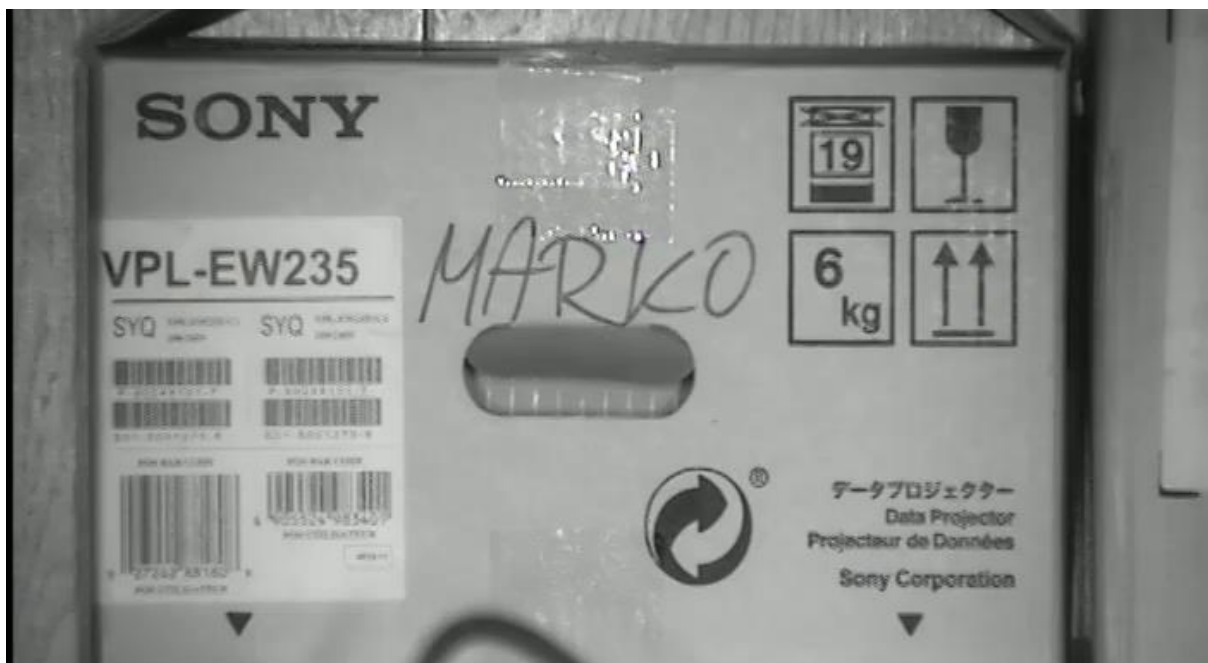
Putem već navedene logike, dobiveno je da je potrebno 7 točaka u horizontalnom i 8 točaka u vertikalnom smjeru da bi se pokrio mogući prostor, odnosno ukupno 56 točaka.



Slika 24: Snimka prostora uz optički zum od 2,5x

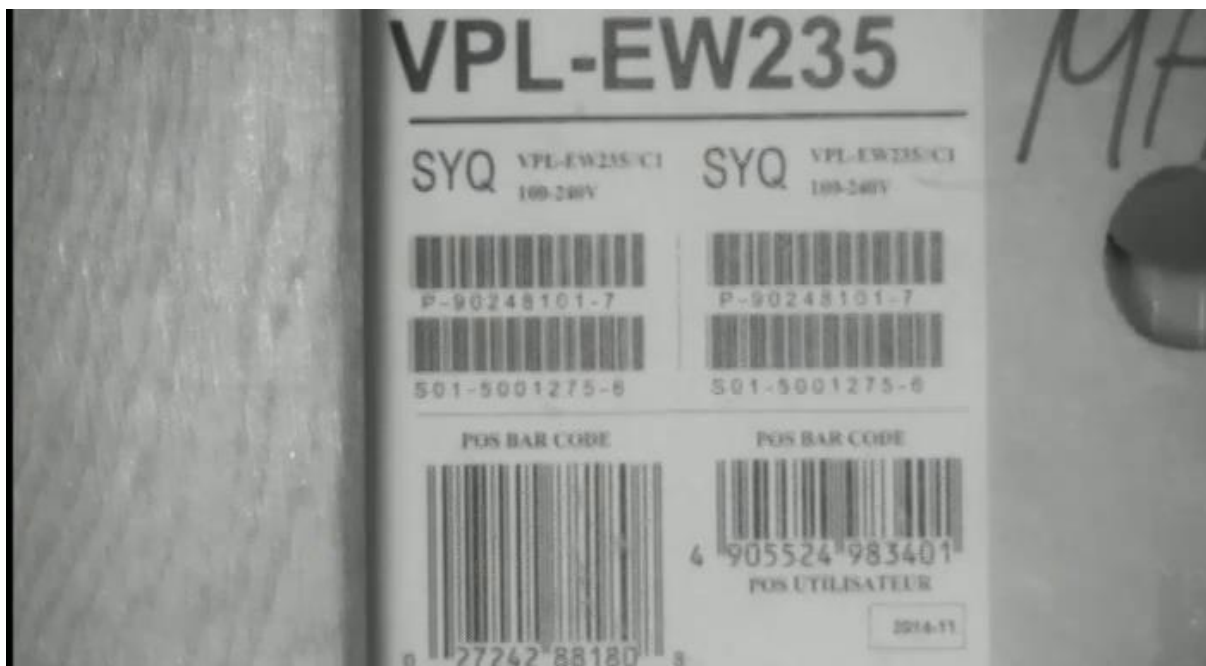
Na slici 2 prikazan je prostor kamerom uz korištenje optičkog zuma od 2,5x s fokusom na pakiranja više proizvoda kao objekata promatranja. Vidljivo je smanjenje kuta, ali i veća razina detalja.

Dobivene vrijednosti su 18 točaka u horizontalnom i 17 točaka u vertikalnom smjeru, što donosi broj od 306 točaka koje su potrebne da bi se pokrio mogući prostor.



Slika 25: Snimka jednog proizvoda uz optički zum od 11x

Na slici 25 prikazano je još veće uvećanje od 11x, gdje je u ovom slučaju fokus na jednom proizvodu. Pripadni kutovi pogleda su još manji nego u prijašnjim primjerima, a za pokrivanje prostora u horizontalnom smjeru potrebno je 76 točaka, dok je za pokrivanje prostora u vertikalnom smjeru potrebno 70 točaka, što donosi broj od čak 5320 točaka potrebnih za pokrivanje cjelokupnog prostora.



Slika 26: Prikaz deklaracije proizvoda uz zum od 20x

U zadnjem slučaju, prikazano je maksimalno uvećanje od 20x, korišteno kako bi se mogla detaljno vidjeti deklaracija proizvoda. Pripadni kutovi pogleda su minimalni, a broj točaka potrebnih za pokrivanje prostora u horizontalnom smjeru iznosi čak 151, dok je za pokrivanje prostora u vertikalnom smjeru potrebno čak 140 točaka. Njihov umnožak iznosi 21 140, što je broj točaka, odnosno kadrova, kojima bi se mogao opisati cjelokupan prostor u okolini kamere. Naravno, lako se može primijetiti da je razina detalja u posljednjem slučaju najveća. Za primjer se može reći da je deklaracija proizvoda na slici veličine 5*10cm, a snimljena je s udaljenosti od otprilike 4 metra. Na prijašnjim slikama nije bilo moguće razaznati deklaraciju s dovoljnom razinom detalja.

6. ZAKLJUČAK

Upotreba *web* servisa te pogotovo standardizacija u vidu SOAP, WSDL, ONVIF i drugih standarda za *web* servise omogućuju vrlo jednostavnu izradu programa za upravljanje uređajima poput korištene IP PTZ kamere, koji bi se, zahvaljujući korištenju široko-prihvaćenih standarda, u istom obliku mogli koristiti i za mnoge druge slične uređaje, tj. kamere, koja bi uključivale podršku za te standarde.

S obzirom na nisku rezoluciju snimanja kamere na kojoj je obavljen eksperimentalni rad, potreban je razmjerno velik broj točaka za visoku preciznost, odnosno potrebna je visoka razina zuma kako bi bilo moguće razaznati detalje. To bi u realnoj primjeni značilo čestu potrebu za promjenom razine zuma kao i zakretanje kamere, a također, radi uskog kuta, odnosno polja gledanja pod visokim zumom, te ovisno o željenoj primjeni, povećanje vremena tokom kojeg kamera može bitno propušta događaje izvan tog dostupnog polja gledanja.

Veća rezolucija kamere omogućila bi smanjenje broja potrebnih točaka nadzora, a istovremeno i veći kut gledanja kod razaznavanja detalja što bi omogućilo bolji nadzor.

7. LITERATURA

- [1] Internet – Wikipedia, <https://en.wikipedia.org/wiki/Internet>
- [2] Brief History of the Internet - Internet Timeline | Internet Society,
<http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet>
- [3] Packet switching – Wikipedia, https://en.wikipedia.org/wiki/Packet_switching
- [4] ARPANET – Wikipedia, <https://en.wikipedia.org/wiki/ARPANET>
- [5] 40 maps that explain the Internet, <https://www.vox.com/a/internet-maps>
- [6] Semantic Web roadmap, <https://www.w3.org/DesignIssues/Semantic.html>
- [7] HTML is Easy, https://en.wikipedia.org/wiki/Apache_HTTP_Server
- [8] XML Introduction, https://www.w3schools.com/xml/xml_what.asp
- [9] XML – Wikipedia, <https://en.wikipedia.org/wiki/XML>
- [10] Cardoso, J., The Syntactic and the Semantic Web,
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.469.1713&rep=rep1&type=pdf>
- [11] Stipančić, T., Semantički Web, predavanja
- [12] Resource Description Framework – Wikipedia,
https://en.wikipedia.org/wiki/Resource_Description_Framework
- [13] SOAP Tutorial, <https://www.tutorialspoint.com/soap/index.htm>
- [14] Overview of SAAJ, https://docs.oracle.com/cd/E17802_01/j2ee/j2ee/1.4/docs/tutorial-update2/doc/SAAJ2.html
- [15] Web Service Definition Language (WSDL), https://www.w3.org/TR/wsdl#_wsdl
- [16] WampServer, la plate-forme de développement Web sous Windows - Apache, MySQL, PHP WampServer, la plate-forme, <http://www.wampserver.com/en/>
- [17] Apache HTTP Server – Wikipedia,
https://en.wikipedia.org/wiki/Apache_HTTP_Server
- [18] PHP: What is PHP? – Manual, <http://php.net/manual/en/intro-what.php>
- [19] PHP: cURL – Manual, <http://php.net/manual/en/book.curl.php>
- [20] PHP syntax and semantics – Wikipedia,
https://en.wikipedia.org/wiki/PHP_syntax_and_semantics
- [21] C# Camera SDK: Introduction to Onvif technology, http://www.camera-sdk.com/p_28-introduction-to-onvif-technology-onvif.html
- [22] SoapUI | Functional Testing for SOAP and REST APIs, <https://www.soapui.org/>

- [23] DIR-600 Wireless N 150 Home Router – D-Link UK,
<http://www.dlink.com/uk/en/support/product/dir-600-wireless-n-150-home-router>

Prilog 1: Izvorni kod test.html stranice

```
<!DOCTYPE HTML>
<html>

<body>
<div id="cctv-container">
    <object classid="clsid:9BE31822-FDAD-461B-AD51-BE1D1C159921"
codebase="http://downloads.videolan.org/pub/videolan/vlc/lates
t/win32/axvlc.cab" id="vlc" events="True" >
        <param name="Src"
value="rtsp://161.53.117.178:554/snl/live/1/1/Ux/sido=-
Ux/sido="/>
        <param name="ShowDisplay" value="True"/>
        <param name="AutoLoop" value="False"/>
        <param name="AutoPlay" value="True"/>
        <embed id="vlcEmb" type="application/x-google-vlc-plugin"
version="VideoLAN.VLCPlugin.2" autoplay="yes" loop="no"
target="rtsp://161.53.117.178:554/snl/live/1/1/Ux/sido=-
Ux/sido=" width="704" height="384"/>
    </object>
</div>
<br>
Speed in % (-100 to 100):
<form action="test.php" method="post">
speed_x: <input type="text" name="speed_x" value = 0
onfocus="if (this.value==0) this.value=''" onblur="if
(this.value=='') this.value=0"><br>
speed_y: <input type="text" name="speed_y" value = 0
onfocus="if (this.value==0) this.value=''" onblur="if
(this.value=='') this.value=0"><br>
speed_zoom: <input type="float" name="speed_zoom" value = 0
onfocus="if (this.value==0) this.value=''" onblur="if
(this.value=='') this.value=0"><br>
```

```
movement_duration: <input type="float"
name="movement_duration" value = 0 onfocus="if (this.value==0)
this.value=''" onblur="if (this.value=='') this.value=0"><br>
<input type="submit" name="test" value="test">
</form>
<br>
<form action="test_preset.php" method="post">
preset: <input type="text" name="preset"><br>
<input type="submit" name="test_preset" value="test preset">
</form>
<br><br>
<form action="test_vlc.php" method="post">
<input type="submit" name="test_vlct" value="test VLC">
</form>

</body>
</html>
```

Prilog 2: Izvorni kod test.php skripte

```
<!DOCTYPE html>
<html>
<body>

<?php
$ch = curl_init();
$headers = [
//      'POST http://192.168.0.200/ HTTP/1.1',
      'POST http://161.53.117.178:10000/ HTTP/1.1',
      'Accept-Encoding: gzip,deflate',
      'Content-Type: application/soap+xml; charset=UTF-
8; action="http://www.onvif.org/ver10/device/wsdl/GetSystemDate
AndTime"',
      'Connection: Keep-Alive',
];

//curl_setopt($ch, CURLOPT_URL, "http://192.168.0.200");
curl_setopt($ch, CURLOPT_URL, "http://161.53.117.178:10000");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

//movement test
$speed_x = $_POST["speed_x"];
$speed_x = $speed_x / 100;
$speed_y = $_POST["speed_y"];
$speed_y = $speed_y / 100;
$speed_zoom = $_POST["speed_zoom"];
$speed_zoom = $speed_zoom / 100;
$movement_duration = $_POST["movement_duration"];
$movement_duration = $movement_duration * 1000000;
```



```

curl_setopt($ch, CURLOPT_POSTFIELDS,
'<soap:Envelope xmlns:sch="http://www.onvif.org/ver10/schema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsdl="http://www.onvif.org/ver20/ptz/wsdl">
    <soap:Header><wsse:Security soap:mustUnderstand="true"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"><wsse:UsernameToken
wsu:Id="UsernameToken-
BBBE5AE173D8C1D5E414939072222924"><wsse:Username>admin</wsse:U
sername><wsse:Password Type="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">admin</wsse:Password><wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-
1.0#Base64Binary">/gkrSsnK+mMs2yFByrTrlQ==</wsse:Nonce><wsu:Cr
eated>2017-05-
04T14:13:42.292Z</wsu:Created></wsse:UsernameToken></wsse:Secu
rity></soap:Header>
    <soap:Body>
        <wsdl:ContinuousMove>

<wsdl:ProfileToken>profile_cam1_stream1</wsdl:ProfileToken>
        <wsdl:Velocity>
            <!--Optional:-->
            <sch:PanTilt x=" ' . $speed_x . ' " y=" ' . $speed_y
. ' "/>
            <!--Optional:-->
            <sch:Zoom x=" ' . $speed_zoom . ' "/>
        </wsdl:Velocity>
        '//<wsdl:Timeout>' . $movement_duration .
'</wsdl:Timeout>
. '    </wsdl:ContinuousMove>

```

```

    </soap:Body>
</soap:Envelope>');
curl_exec($ch);

// stop test
usleep($movement_duration);
curl_setopt($ch, CURLOPT_POSTFIELDS,
'<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-
envelope" xmlns:wsdl="http://www.onvif.org/ver20/ptz/wsdl">
    <soap:Header><wsse:Security soap:mustUnderstand="true"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"><wsse:UsernameToken
wsu:Id="UsernameToken-
BBBE5AE173D8C1D5E414939073949846"><wsse:Username>admin</wsse:U
sername><wsse:Password Type="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">admin</wsse:Password><wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-
1.0#Base64Binary">eGZ/cFwqyexpM2ipW8c5DQ==</wsse:Nonce><wsu:Cr
eated>2017-05-
04T14:16:34.983Z</wsu:Created></wsse:UsernameToken></wsse:Secu
rity></soap:Header>
    <soap:Body>
        <wsdl:Stop>

<wsdl:ProfileToken>profile_cam1_stream1</wsdl:ProfileToken>
        <!--Optional:-->
        <wsdl:PanTilt>0</wsdl:PanTilt>
        <!--Optional:-->
        <wsdl:Zoom>0</wsdl:Zoom>
    </wsdl:Stop>

```

```
</soap:Body>
</soap:Envelope>');
curl_exec($ch);
curl_close ($ch);

header("Refresh: 0, url=test.html");
?>

</body>
</html>
```

Prilog 3: Izvorni kod test_preset.php skripte

```
<!DOCTYPE html>
<html>
<body>

<?php
$ch = curl_init();
$headers = [
//      'POST http://192.168.0.200/ HTTP/1.1',
      'POST http://161.53.117.178:10000/ HTTP/1.1',
      'Accept-Encoding: gzip,deflate',
      'Content-Type: application/soap+xml;charset=UTF-
8;action="http://www.onvif.org/ver10/device/wsdl/GetSystemDate
AndTime"',
      'Connection: Keep-Alive',
];

//curl_setopt($ch, CURLOPT_URL,"http://192.168.0.200");
curl_setopt($ch, CURLOPT_URL,"http://161.53.117.178:10000");
curl_setopt($ch, CURLOPT_POST, 1);

//preset test
curl_setopt($ch, CURLOPT_POSTFIELDS,
'<soap:Envelope xmlns:sch="http://www.onvif.org/ver10/schema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsdl="http://www.onvif.org/ver20/ptz/wsdl">
    <soap:Header><wsse:Security soap:mustUnderstand="true"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"><wsse:UsernameToken
wsu:Id="UsernameToken-
BBBE5AE173D8C1D5E414939093288358"><wsse:Username>admin</wsse:U
```

```

    sername><wsse:Password Type="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">admin</wsse:Password><wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-
1.0#Base64Binary">/eg7j/IzeCV3SRsgkdoENQ==</wsse:Nonce><wsu:Cr
eated>2017-05-
04T14:48:48.835Z</wsu:Created></wsse:UsernameToken></wsse:Secu
rity></soap:Header>
    <soap:Body>
        <wsdl:GotoPreset>

<wsdl:ProfileToken>profile_cam1_stream1</wsdl:ProfileToken>
        <wsdl:PresetToken>preset' . $_POST["preset"] .
'</wsdl:PresetToken>
        </wsdl:GotoPreset>
    </soap:Body>
</soap:Envelope>');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

curl_exec($ch);
curl_close ($ch);
header("Refresh: 0, url=test.html");
?>

</body>
</html>

```

Prilog 4: Izvorni kod test_vlc.php skripte

```
<!DOCTYPE html>
<html>
<body>

<?php
require __DIR__ . '/vendor/tesseract-ocr-for-php-
master/src/TesseractOCR.php';
$time = date('Y.m.d_H.i.s');
//$vlc = 'C:\Progra~2\VideoLAN\VLC\vlc.exe
rtsp://192.168.0.200:554/snl/live/1/1/Ux/sido=-Ux/sido= --
rate=1 --video-filter=scene --vout=dummy --scene-format=png --
scene-ratio=12 --scene-prefix=snap --scene-
path=C:\wamp64\www\test_vlc\' . $time . '\\ --run-time=2
vlc://quit';
$vlc = 'C:\Progra~2\VideoLAN\VLC\vlc.exe
rtsp://161.53.117.178:554/snl/live/1/1/Ux/sido=-Ux/sido= --
rate=1 --video-filter=scene --vout=dummy --scene-format=png --
scene-ratio=12 --scene-prefix=snap --scene-
path=C:\wamp64\www\test_vlc\' . $time . '\\ --run-time=2
vlc://quit';
echo("<a href=\""/test_vlc/" . $time . "\">File list</a>");
echo('<br><br>');
exec('mkdir C:\wamp64\www\test_vlc\' . $time);
exec($vlc);
//header("Refresh: 0, url=test.html");
echo("<img src=\""/test_vlc/" . $time . "/snap00001.png\"/>");
echo (new TesseractOCR("/test_vlc" . $time .
"/snap00001.png"))
    ->psm(11)
    ->run();

echo('<br><br>');
```

?>

</body>

</html>